HEXA-X-II

A holistic flagship towards the 6G network platform and system, to inspire digital transformation, for the world to act together in meeting needs in society and ecosystems with novel 6G services

Deliverable D6.5
Final Design on 6G Smart Network Management Framework

Co-funded by the European Union

6GSNS

Hexa-X-II project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101095759.

Date of delivery:	28/02/2025	Version:	1.0
Project reference:	101095759	Call:	HORIZON-JU-SNS-2022
Start date of project:	01/01/2023	Duration:	30 months

Document properties:

Document Number:	D6.5
Document Title:	Final Design on 6G Smart Network Management Framework
Editor(s):	A. Zafeiropoulos, I. Tzanettis, G. Kakkavas, S. Papavassiliou (ICC)
Authors:	I. Labrador Pavón (ASA), G. Landi, P.G. Giardina, E. Seder (NXW), J.M. Jorquera Valero, M. Gil Pérez (UMU), C. Ayimba (UC3), Ricard Vilalta, Lluis Gifre, Pol Alemany, Behnam Ojaghi, Raul Muñoz (CTT), Toni Dimitrovski, Nassima Toumi (TNO), Jakob Miserez, Wouter Tavernier (IMEC), Santiago Rodriguez, José Cunha, Fernando Lamela, Xosé Ramón Sousa (OPT), Vasiliki Lamprousi, Sokratis Barmpounakis, Panagiotis Demestichas (WIN), J. Cáceres Galán, E. Lluesma Martí (ATO), Elham Dehghan Biyar, Mehmet Karaca (EBY), Flávio Brito, Franco Ruggeri, Josue Castaneda Cisneros (EAB), Riccardo Nicolicchia (TID)
Contractual Date of Delivery:	28/02/2025
Dissemination level:	PU
Status:	Final
Version:	1.0
File Name:	Hexa-X-II D6.5_v1.0

Revision History

Revision	Date	Issued by	Description
0.1	20.06.2024	Hexa-X-II WP6	Initial ToC
0.2	28.06.2024	Hexa-X-II WP6	Updated ToC
0.3	28.07.2024	Hexa-X-II WP6	Initial contributions in Section 2.1
0.4	30.09.2024	Hexa-X-II WP6	Finalisation of Section 2.1 and contributions in Section 2.2
0.5	15.11.2024	Hexa-X-II WP6	Internal review of Section 2 and finalisation of contributions in Section 2.2
0.6	29.11.2024	Hexa-X-II WP6	Contributions in Section 3 and 4
0.7	11.12.2024	Hexa-X-II WP6	Internal review provided
0.8	20.12.2024	Hexa-X-II WP6	Revised version based on the comments from the internal review
0.9	27.01.2025	Hexa-X-II WP6	Revised version based on the comments from the external review
1.0	30.01.2025	Hexa-X-II WP6	Final version

Abstract

This document describes the design and implementation of the 6G Smart Network Management Framework, contributing to the overall Hexa-X-II end-to-end system blueprint. The deliverable provides an overview of the Smart Network Management Framework defined in the project, considering the specification of various architectural Management and Orchestration (M&O) solutions, functionalities, specific systems and algorithms. The document provides implementation details, both for individual components of the 6G Smart Network Management Framework, as well as implementations based on workflows that consider the interaction among multiple components of it. Per case, a set of workflows are provided, accompanied by evaluation results related to the proposed functionalities. Then, information is provided related to the alignment of the Smart Network Management Framework with the overall HEXA-X-II project activities. This includes the interaction of WP6 with the other Work Packages and the contribution to the Hexa-X-II objectives, the performed dissemination and standardisation activities, the contribution to the project's Key Exploitable Results, the alignment with the project use cases, the impacted Key Performance Indicators (KPIs) and Key Value Indicators (KVIs), and the alignment with the recommendations provided by the Advisory Group.

Keywords

6G Smart Network Management Framework, Management and Orchestration, Monitoring, Data Fusion, Control Loops

Executive Summary

The Smart Network and Services Joint Undertaking (SNS JU) 6G Flagship project Hexa-X-II leads the way to next generation 6G end-to-end (E2E) system design, based on integrated and interacting technology enablers. The project will continue the track of the previous 5G-PPP Hexa-X project [HEX24], which has laid the foundation for the global communication network of the 2030s by developing the 6G vision and basic concepts. As a continuation of Hexa-X, the interaction between the cyber-physical world and human world will be further evolved with the advancement of information, communication, and computation technologies towards a pervasive human centred cyber-physical world in 2030. To reach this 6G vision, Hexa-X-II has set up the goals to design the system blueprint of a sustainable, inclusive, and trustworthy 6G platform, which will require novel enablers regarding Smart Management & Orchestration (M&O) functionalities.

In this direction, this document is the final public deliverable produced by the Hexa-X-II "Smart Network Management" Work Package (WP6), documenting the evolution in the design and implementation of the envisioned 6G Smart Network Management Framework and a set of evaluation results. WP6 has defined a framework considering: (i) the contributions to the Hexa-X-II architecture design principles, (ii) the mapping with the envisaged 6G stakeholders, (iii) the definition and the alignment of the WP6 M&O technical enablers with the initial blueprint provided from WP2, and (iv), the envisaged contributions of those M&O enablers towards the future 6G smart networks, as detailed in D6.3 [HEX224-D63]. The primary goal of this framework is to serve as a reference structure for other Work Packages (WPs) in the project, and mainly to WP2, to support the design and implementation of the final end-to-end (E2E) system blueprint addressed in such WP. However, the ambition is of course that this WP6 framework can be used as a reference beyond the Hexa-X-II project itself, also becoming a referent for designing and implementing the actual management and orchestration (M&O) systems of the future 6G networks. The WP6 framework is divided into four main sections:

- The Overall Architectural M&O Solutions, which contains the systems designed to manage and orchestrate resources and services across the entire network continuum, enabling the M&O of resources and services even beyond the technical and administrative boundaries of individual stakeholders.
- The Specific Systems that would be deployed in the scope of specific stakeholders, e.g., network resources programmability systems, provisioning systems, or others.
- The Overall Functionalities that are more specific in scope than the Overall Architectural M&O Solutions. Such functionalities include, for instance, the monitoring functionality or the trust management systems.
- The Algorithms that would be deployed on the stakeholder's scope. A set of algorithms is provided that were found of interest in WP6, in line with the Hexa-X-II project work programme.

For each section of the framework, details are provided for the designed and developed components. Initially, the design of each component is provided, focusing on the main supported characteristics and its suitability for managing specific functionalities or services of the framework. Following, implementation details are given, considering both the implementation of individual components of the framework, as well as implementation of workflows that engage multiple components of the framework. Evaluation results are presented in most of the cases based on the status of evaluation activities in the project. For part of the cases, the provided evaluation results are considered final, while for the rest, evaluation activities are going to be continued in the context of the PoC developments in WP2 towards the end of the lifetime of the Hexa-X-II project work programme.

Furthermore, the document provides information regarding the alignment of the activities in WP6 with the Hexa-X-II project work programme. Details are presented for the interaction among WP6 and the other Hexa-X-II WPs, the dissemination and standardisation activities performed in the context of WP6, the WP6 contribution to the project Key Exploitable Results, the alignment of the work in WP6 with the project use cases, the estimated impact on the Key Performance Indicators (KPIs) and Key Value Indicators (KVIs) defined in the Hexa-X-II project, the contribution from WP6 to the Hexa-X-II project objectives, and the alignment with the recommendations provided by the Advisory Group.

Finally, a set of concluding remarks are detailed for the work done within WP6 in the Hexa-X-II project.

Table of Contents

List of Tables

List of Figures

Acronyms and abbreviations

Term	Description
ACO	Ant Colony Optimisation
AF	Analytics Function
AI	Artificial Intelligence
AMR	Autonomous Mobile Robot
ANFIS	Adaptive Neuro Fuzzy Inference System
API	Application Programming Interface
CI/CD	Continuous Integration & Continuous Delivery
CL	Closed Loop
CLI	Command Line Interface
DQN	Deep Q-Network
DetNet	Deterministic Networking
DLT	Distributed Ledger Technologies
DRL	Deep Reinforcement Learning
ETSI	European Telecommunications Standards Institute
E2E	End-To-End
ERAB	E-UTRAN Radio Access Bearer
FL	Federated Learning
IBN	Intent-based Networking
GenAI	Generative Artificial Intelligence
GUI	Graphical User Interface
GNN	Graph Neural Network
IbM	Intent-based Management

ICT	Information and Communications Technology
ILE	Infrastructure Layer Emulator
IMF	Intent Management Function
INT	In-Band Network Telemetry
INT-MD	In-Band Network Telemetry - eMbed Data
INT-MX	In-Band Network Telemetry - eMbed instruct(X)ions
IoT	Internet of Things
ISG	Industry Specification Group
ISPM	Infrastructure Status Prediction Module
KER	Key Exploitable Result
KPI	Key Performance Indicator
KVI	Key Value Indicator
K8s	Kubernetes
LoT	Level of Trust
LoTAF	Level of Trust Assessment Function
MAE	Mean Absolute Error
MARL	Multi-Agent Reinforcement Learning
MBR	Maximum Bit Rate
MCE	Management Capabilities Exposure
MDAF	Management Data Analytics Function
M&O	Management and Orchestration
ML	Machine Learning
MLOps	Machine Learning Operations
MNO	Mobile Network Operator
MSE	Mean Square Error
NB-IoT	Narrowband Internet of things
NDT	Network Digital Twin
NEF	Network Exposure Function
NER	Named Entity Recognition
NF	Network Function
NSMF	Network Slice Management Function
NWDAF	Network Data Analytics Function
OTLP	Open Telemetry Protocol
PMF	Privacy Management Function

PoC	Proof of Concept
POF	Privacy Operation Function
QoE	Quality of Experience
QoS	Quality of Service
QT	Quantifiable Target
RAN	Radio Access Network
RBAC	Role-Based Access Control
RL	Reinforcement Learning
RT	Real-Time
RTT	Round-Trip Time
RRC	Radio Resource Control
SAIN	Service Assurance for Intent-Based Networking
SBMA	Service-based Management Architecture
SDN	Software-Defined Networking
SDG	Sustainability Development Goal
SC	Smart Contract
S-MLOps	Sustainable Machine Learning Operations
SLA	Service Level Agreement
SLO	Service Level Objective
SV	Software Vendor
TAS	Time Aware Shaping
TFS	TeraFlowSDN
TEF	Trust Evaluation Function
TMS	Trust Management System
TLA	Trust Level Agreement
TLS	Transport Layer Security
TSN	Time-Sensitive Networking
UCSPS	User-Centric Service Provisioning System
URLLC	Ultra Reliable Low Latency Communication
URSP	User Equipment Route Selection Policy
UAV	Unmanned Aerial Vehicle
UE	User Equipment
Wi-Fi	Wireless Fidelity
ZSM	Zero-touch network and Service Management

1. Introduction

1.1 Objective of the document

This document describes the design and implementation of the 6G Smart Network Management Framework, contributing to the overall Hexa-X-II end-to-end system blueprint. It regards the final deliverable of the WP6 of the Hexa-X-II project work programme.

The document provides an overview of the Smart Network Management Framework defined in the project, considering the specification of various architectural Management and Orchestration (M&O) solutions, functionalities, specific systems, and algorithms. The document provides design and implementation details, both for individual components of the 6G Smart Network Management Framework, as well as implementations based on workflows that consider the interaction among multiple components of it. Per case, a set of workflows are provided, accompanied by evaluation results related to the proposed functionalities. Part of the provided evaluation results regard work in progress, considering that some implementations are evolving in the framework of the development and evaluation of the work done in the PoCs, as managed by WP2. The final version of the evaluation results will be made available in D2.6.

Following, information is provided related to the alignment of the Smart Network Management Framework with the overall HEXA-X-II project activities. This includes the interaction of WP6 with the other Work Packages and the contribution to the Hexa-X-II objectives, the performed dissemination and standardisation activities, the contribution to the project's Key Exploitable Results, the alignment with the project use cases, the impacted Key Performance Indicators (KPIs) and Key Value Indicators (KVIs), and the alignment with the recommendations provided by the Advisory Group.

Finally, updated information for some of the enablers detailed in D6.3 [HEX224-D63] or added in the last phase of the WP6 activities is provided in the annexes of the document (Section 6). These enablers regard the Level of Trust Assessment Function, the Privacy Protection Framework for Data Analytics in M&O, the Open Telemetry and Data Fusion, the Functionality Allocation and Physical Task Planning, and the Real-Time (RT) zero-touch Cognitive Closed-loop Automation.

1.2 Structure of the document

The document is structured as follows:

- Section 1 introduces the document with an overview of its content and structure, defining the main objectives of the deliverable together with its position and main contribution to the Hexa-X-II project.
- Section 2 details the Smart Network Management Framework based on the designed overall architectural M&O solutions, overall functionalities, specific systems and algorithms. Implementation details and evaluation results are provided.
- Section 3 details the alignment of the Smart Network Management Framework with the overall HEXA-X-II project activities.
- Section 4 concludes the document, providing information for the main outcomes of the WP6 activities and their adoption in the various WPs in the HEXA-X-II project work programme.
- The Appendices provide additional details on enablers that were developed or updated the last working period, since the release of the D6.3 [HEX224-D63], as well as on the full list of Key Performance Indicators (KPIs) and the metrics for their assessment for the various components of the Smart Network Management Framework.

1.3 Terminology

For the purposes of the present document, the terms in the following documents apply:

- 3GPP TR 21.905 – V18.0.0 – Vocabulary for 3GPP Specifications [3GP21].
- ETSI GR NFV 003 - V1.6.1 - Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV [ETS21].

- International Telecommunication Union (ITU), Maintenance: Introduction and General Principles of Maintenance and Maintenance Organisation – Maintenance Terminology and Definitions, ITU-T Recommendation M.60. March 1993 [ITU93].

Relevant concepts related to our work in these documents:

Core network: An architectural term relating to the part of 3GPP System which is independent of the connection technology of the terminal (e.g., radio, wired) [3GP21].

Orchestration: Orchestration is sequencing where a management operation is dependent upon several managed objects in a network being changed in a strict sequence [ITU93].

Service: component of the portfolio of choices offered by service providers to a user, a functionality offered to a user. NOTE 1: As defined in ETSI TR 121 905 [3GP21]. NOTE 2: A user can be an end-customer, a network or some intermediate entity [ETS21].

Resource: Manageable functional parts of telecommunication and support equipment which can be unambiguously defined [ITU93].

Network Service (NS): composition of Network Function(s) and/or Network Service(s), defined by its functional and behavioural specification [ETS21].

Service Enabler: a capability which may be used, either by itself or in conjunction with other service enablers, to provide a service to the end user [3GP21].

Application: an application is a service enabler deployed by service providers, manufacturers or users. Individual applications will often be enablers for a wide range of services. (UMTS Forum report #2) [3GP21].

Additional definitions not in the previous references:

- **Network continuum:**

  Heterogeneous set of network infrastructure resources (i.e., interconnect, compute, and storage resources), physical or virtualised, spanning across different administrative and technological domains, but exposed to stakeholders as a single integrated resource. Also known as device-edge-cloud continuum or cloud continuum.

- **Continuum computing:**

  Any type of computing performed on the network continuum.

- **Continuum orchestration:**

  Continuum computing targeting the orchestration as defined in [ITU93], of the network continuum itself (aka resources continuum orchestration) and those network services that may be deployed on it (aka network services continuum orchestration). It can be also referred as "meta-orchestration", where the "meta-" prefix refers to its extension to all those infrastructure resources and service components that may be outside the limits of the common orchestration concept, as well as to the shift in the common orchestration paradigm.

- **Edge Computing:**

  A concept, as described in 3GPP TS 23.501 [23.501], that enables operator and 3rd party services to be hosted close to the User Equipment (UE) access point of attachment, to achieve an efficient service delivery through the reduced end-to-end latency and load on the transport network.

- **Edge Network:**

  A network that supports edge computing.

- **Extreme-edge:**

Those resources in the network continuum beyond the administrative and technical domains of a specific stakeholder, also part of that network continuum. It may include UEs, other Customer Premise Equipment's (CPEs), Internet of Things (IoT) devices, or external -to that stakeholder- private or public networks, among others. Infrastructure resources in this domain can be highly heterogeneous, mobile, volatile, and belonging to a multiplicity of stakeholders. The extreme-edge can also be massive in scale. Computing performed on this extreme-edge domain can be referred to as "extreme-edge computing".

2. Smart Network Management Framework

This section describes the main topic in this document, the Smart Network Management Framework. Initially, Section 2.1 presents the overall design of the framework, describing its overall structure and building blocks. These blocks are basically an evolution of the technical enablers identified in the previous deliverables in this WP6, i.e., [HEX223-D62] and [HEX224-D63], but arranged in a more integrated and structured way, and provided with updated descriptions clarifying their role and scope within the overall framework design. For those components of the framework in which further progress has been made since the release of the previous deliverable [HEX224-D63], additional details are also provided in the annexes (Section 6). Following the overall framework description, Section 2.2 focuses on the related implementations, with two main subsections: Section 2.2.1, describing implementations of certain components of the framework, and Section 2.2.2, describing implementations based on the framework, i.e., targeting to showcase how the framework design presented here could be used in practice. After that, Section 2.2.3 presents certain contributions made to the software developer's community in relation to the implementations presented in the previous section. Finally, Section 2.2.4 also presents the main supporting technologies that have been used for these implementations, considering both those specifically developed in the context of this Hexa-X-II project, as well as other external state-of-the-art technologies.

2.1 Framework Design

This section provides a comprehensive description of the 6G Smart Network Management Framework as a whole. The primary goal of this framework is to serve as a reference structure for other Work Packages (WPs) in the project, and mainly to WP2, to support the design and implementation of the final end-to-end (E2E) system blueprint addressed in such WP. However, the ambition is of course that this WP6 framework can be used as a reference beyond the Hexa-X-II project itself, also becoming a referent for designing and implementing the actual management and orchestration (M&O) systems of the future 6G networks.

Figure 2-1 shows the overall diagram of this Smart Network Management Framework. As it can be appreciated the framework integrates those functionalities commonly integrated in the M&O systems, such as those associated with the services life-cycle management mechanisms (e.g., service provisioning and assurance), monitoring, configuration management, resource management, or security functionalities, among others. These mechanisms are represented by the list of coloured numbers in the middle of the figure, which, as it can be seen, are assigned to each of the components of the framework depending on the main functionalities to which they are associated. This mapping between M&O-related functionalities and components considers the main functionalities envisaged for each component, regardless of the possibility that each component may perform other functionalities, or that it may work together with other components in the framework to perform other functionalities as well.

More specifically, these numbers refer to:

1. Network Services Provisioning (and de-provisioning), considering the network services as a whole, or their building components.
2. Network Services Assurance, referring to the mechanisms to ensure that deployed network services always behave according to the required performance and Quality of Service (QoS) levels. This category also includes service failures management, and Service Level Agreement (SLA) fulfilment.
3. Network Services Configuration, including operating policies configuration.

4. Network Resources Provisioning (and de-provisioning), specifically targeting the provisioning of softwarised infrastructure network resources (servers, storage resources, or networking resources)[1].
5. Network Resources Assurance, which refers the processes to keep softwarised network infrastructure resources up and running. It also includes failures management.
6. Network Resources Configuration, referring the configuration of the physical or the softwarised infrastructure network resources.
7. Monitoring and Analytics (for both services and resources), including also the QoS and policies monitoring, as well as alerts, alarms, and reporting generation.
8. Security (for both, services and resources). It includes intrusion detection, identity and access management, and threat protection.
9. Design and Development related systems.
10. Capabilities Access, referring the assets or functionalities to make it possible for external entities (to a certain stakeholder) to interact with and utilise specific capabilities or resources within the M&O system.
11. Privacy, referring the secure handling of sensitive user and network data, and protect against unauthorised access, support compliance with regulations, and uphold user confidentiality by anonymizing, encrypting, or restricting data based on privacy policies.

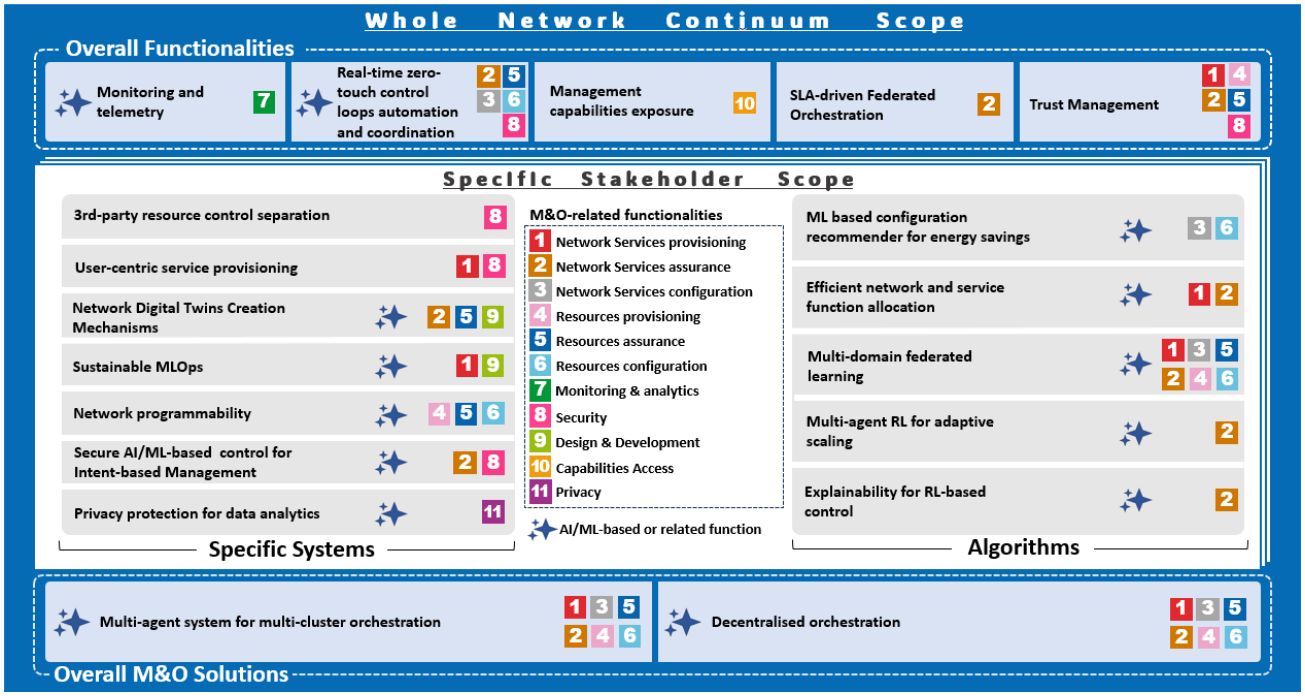

Figure 2-1: Smart Network Management Framework.

However, beyond these common M&O functionalities, the framework also integrates cutting edge technologies, which are used to implement these common M&O functionalities, including the extensive use of Artificial Intelligence (AI) and Machine Learning (ML) mechanisms (explicitly represented in the figure by the sparkle icon - ✨) to enable advanced 6G network management solutions, and which indeed allows to qualify this management framework as "smart".

Also, as it can be appreciated, the figure categorises the different building blocks according to their scope of application into two large groups:

a) Those enablers that apply to the *entire* network continuum. These are the blocks depicted in the outer dark blue frame in the figure.

[1] Please note that, in what regards this document, the terms "resource" and "service", are referring those specific concepts defined in the previous Section 1.3.

b) Those enablers that would be deployed on certain specific stakeholders' part of that continuum. These are the light grey blocks in the white rectangle enclosed by the blue frame.

This classification is in line with the multi-stakeholder concept considered in the E2E system design in WP2 towards 6G [HEX224-D23], which considers a multiplicity of stakeholders, such as network operators, vertical industries, hyper-scalers, or neutral-hosts, among others, and all of them being part of the network continuum (see [HEX224-D63] – Section 2.3). In fact, the representation of the specific stakeholder's scope "within" the whole network continuum in the framework figure is intentional, representing that the individual stakeholders are "part of" such continuum.

But as it can be seen in Figure 2-1, beyond this general scope-based categorisation, the WP6 framework is also divided into four main sections:

1. The so-called **Overall M&O Solutions** block (bottom), which contains the systems designed to manage and orchestrate resources and services across the entire network continuum, i.e., extending the M&O capabilities across the various technical and administrative domains of the network, from a specific stakeholder's core network to its extreme-edge, thus enabling the M&O of resources and services even beyond the technical and administrative boundaries of individual stakeholders. As it can be appreciated, these Overall M&O Solutions are general M&O systems including most of the mechanisms in the numbering list in the middle of the figure.
2. The long rectangle on top also covers the whole network continuum, but in this case, focusing on the so-called **Overall Functionalities**, like for instance, the monitoring functionality or the trust management systems in the figure. As it can be appreciated, these Overall Functionalities are more specific in scope than the Overall Architectural M&O Solutions described before.
3. **Specific Systems**. This is the set of blocks on the left side of the figure, in the Specific Stakeholders block. It represents those systems which are not overall, i.e.,| that would be deployed in the scope of specific stakeholders targeting specific purposes in their own scope, such as network resources programmability systems, provisioning systems, or others.
4. Finally, the set of blocks on the right side of the figure represents those **Algorithms** that would be deployed on the stakeholder's scope. Of course, this is not a complete list of all the algorithms that would be deployed on a final 6G system; it just shows those algorithms in the WP6 scope, in line with the Hexa-X-II project work programme[2].

In the following subsections each of these blocks are described in more detail[3].

2.1.1 Overall Architectural M&O Solutions

This section outlines two distinct approaches to M&O for the entire network continuum: a hierarchical approach and a decentralised approach.

Both approaches offer general M&O solutions targeting the provisioning, assurance, and configuration of network services and resources over the whole network continuum, i.e., from the stakeholder's core network to the extreme-edge domain, and in a highly distributed way. Both approaches consider the multi-stakeholder

[2] Please note that the framework does not explicitly include a block for Intent-based Management, which however, is considered a relevant topic in what regards the management of the future 6G networks. This is because, according to the Hexa-X-II work programme, this functionality is not addressed in WP6, but in WP2. However, it is considered that for implementations beyond the Hexa-X-II project, this Intent-based Management functionality should also be considered part of the M&O framework, both as a functionality encompassing the orchestration mechanisms over the entire network continuum, and also, to implement certain stakeholder-specific orchestration mechanisms. Anyway, beyond this, and as it can be appreciated, the WP6 management framework includes some specific functionalities associated to that intent-based management topic addressed in WP2.

[3] Since most of the technical enablers described in these sections were already introduced in the previous Deliverable 6.3 [HEX224-D63], only a summary description of these enablers is included in this subsection 2.1 for the convenience of the reader. For those enablers that were not described in D6.3, or for those that, although already introduced, have been further developed, although a summary is also provide in this section 2.1, more detailed information can also be found in the annexes (Section 6).

environment envisaged towards the future 6G networks and the challenging features of the extreme-edge domain, i.e., the domain beyond a specific stakeholder own domain with highly volatile resources, cloud-native scalability [OCM09], and with a wide range of devices. The latter include not just regular computing and storage resources belonging to other stakeholders, but also, non-reliable CEP (Consumer Equipment Premises) devices, such as IoT devices or even small-scale battery-powered devices.

Implementing this concept goes beyond merely connecting various devices outside a specific stakeholder's network (as it is already common in the state-of-the-art, e.g., by integrating IoT or end-user devices for data collection). It also involves enabling the deployment and the management of network services and resources on the "beyond the edge" infrastructure in a cloud-native manner. The key idea is that the combined computing power and storage capacity of extreme-edge resources can be used to distribute workloads more efficiently, decrease data communication needs, and when very close to end-users, substantially contribute to reduce latency that could go beyond the figures in the current 5G technology. Also, from a business perspective, the usage of resources beyond the own network can help certain stakeholders (e.g., mobile network operators) to implement new profitable business models [Law24].

The decentralised approach is designed to manage the entire network ecosystem in a highly distributed manner, covering everything from the core network of stakeholders to the extreme-edge. This approach introduces a new set of distributed network elements with the primary goal of ensuring service continuity, scalability, and optimised resource utilisation [HEX224-D33]. It is particularly focused on supporting multi-stakeholder environments and volatile resources at the extreme edge, including also AI/ML resources to support the management of this scope. It considers that the extreme-edge domain can scale to a massive size, in line with cloud-native principles, and encompassing a wide variety of devices, including not only traditional computing and storage resources. The decentralised M&O system is intended to orchestrate network services on these diverse resources, ensuring effective operation in this heterogeneous environment.

On the other hand, the hierarchical approach leverages a multi-agent system for multi-cluster orchestration. It relies on a hierarchical decision-making model, where a centralised E2E orchestrator manages resources and services across multiple platforms and administrative domains, including also the extreme-edge. This model also integrates AI/ML-based techniques for the allocation of service components and enables proactive M&O actions based on predictive insights. The centralised orchestrator plays a key role in optimizing the allocation of resources and managing network services efficiently.

Both orchestration approaches are offered in the framework to give stakeholders the possibility to select the approach that best fits their design requirements. In the following subsections, both solutions are described in more detail.

2.1.1.1 Multi-agent system for multi-cluster orchestration

Multi-agent systems refer to self-organised systems that are based on the interactions among multiple agents to provide solutions and solve problems that cannot be easily solved by an individual agent [DKJ18]. They consist of a set of agents and the environment over which they operate. Collaboration and establishment of synergies among the agents is crucial to achieve their goal. Each agent has its local view of the environment, while it contributes to the development of a global view based on the exchange of information among agents. Decision-making can be supported in local and global level depending on the problem to be solved and the posed constraints.

Multi-agent systems are considered a good solution for the development of AI/ML-driven orchestration mechanisms for the computing continuum [KLM22] (e.g. the Multi-Agent Reinforcement Learning paradigm [ZFF+24]), where multiple clusters of infrastructure are made available for the deployment of applications and services, ranging from the extreme edge to the edge to the cloud part of the infrastructure (Figure 2-2). Different type of orchestration mechanisms can be developed, introducing distributed and/or decentralised AI-based techniques for the end-to-end management of network services and applications. Two main aspects are considered: the need to address management of heterogeneous resources that are made available across distributed clusters in the continuum and the need to increase the intelligence and automation characteristics of the provided orchestration mechanisms considering both, the deployment and operational/runtime phases of network services and applications.

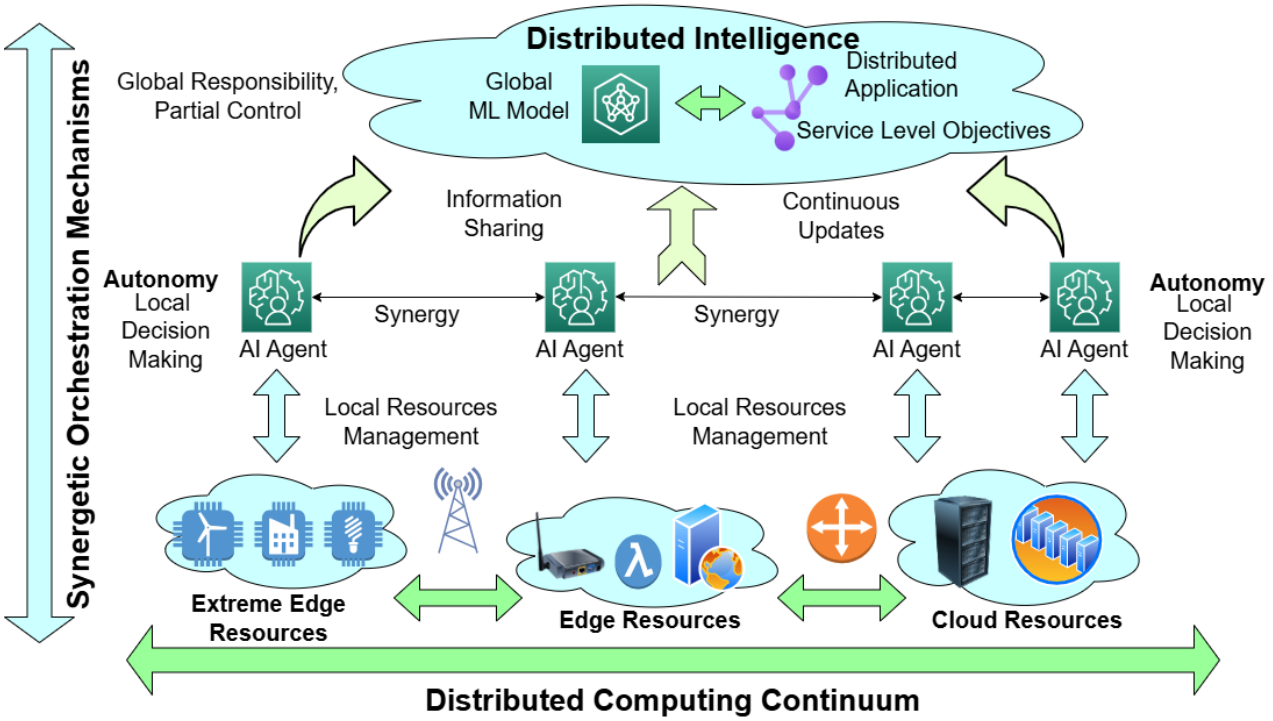

Figure 2-2: High level view of a multi-agent management approach for the computing continuum [ZFF+24]

Various levels of synergy between resource management systems are considered. Synergies may be applied through the adoption of multi-agent systems that collaborate to achieve a joint objective. Multi-agent systems are going to be used for the development of orchestration mechanisms supporting varied scopes, ranging from high-level, single-agent management per cluster to more granular configurations with individual agents managing each microservice or computing or network node. Such mechanisms can support various actions, including the autoscaling of functions/microservices of a specific service graph, deployment and compute offloading techniques with specific optimisation objectives, and migration mechanisms. Synergies may be also applied among different stakeholders in a 6G ecosystem. For instance, interaction among network providers and over the top (OTT) players such as edge/cloud application providers could be applied. In this case, the specification and development of open APIs is envisaged, mainly northbound interfaces offered by network providers to OTT players.

For supporting the envisaged requirements in 6G systems implemented by joint network and compute infrastructures, there is a need to identify the different kinds of stakeholders developing concerns in such ecosystems [YMP23]. To do so, we are based on the main roles envisaged in a 6G system, as detailed in [HEX224-D63]. These roles include the “Digital Service Provider”, the “Network Operator”, and the “Virtual Infrastructure Service Provider”. Digital Service Providers are responsible to provide their services to end users and can be considered as the customers of the ecosystem with specific requirements and intents to satisfy. The Network Operators are responsible to manage the deployment and provision of the service, taking into account their operational policies and the requirements provided by the Digital Service Provider. The deployment is done over programmable resources offered by Virtual Infrastructure Service Providers. Such resources are virtual resources that span across the extreme edge, edge and cloud part of the infrastructure. Depending on the owner of the service and the infrastructure, such roles can be aggregated (e.g., in case that the Network Operator deploys services over its own infrastructure, it also undertakes the role of the Virtual Infrastructure Service Provider).

The following architectural approach (Figure 2-3) is suggested for supporting multi-agent decision-making, multi-stakeholder interplay and multi-domain service federation. Based on the specification of an intent for a service deployment, the Multi-agent Decision Support System is responsible for the lifecycle management of the service (deployment, re-configuration, deprovision). Depending on the available resources and the network

topology, multiple agents may be activated to manage the various orchestration mechanisms across the programmable infrastructure. Such agents may take up to generate and enforce automated management actions such as scaling, migration, service offloading, bandwidth allocation etc. A multi-cluster Deployment Manager undertakes the production of a deployment plan and the provision of the service over compute and network resources, where an agent is responsible for local orchestration actions per cluster, service or network link. In cases where multiple administrative domains are considered in the deployment, the Federation Manager is responsible to support the negotiations among agents. Multi-domain observability and analysis mechanisms are considered as enabling technologies to assist the decision-making through the production of knowledge and insights. Following, short information is provided per component of the provided architectural approach.

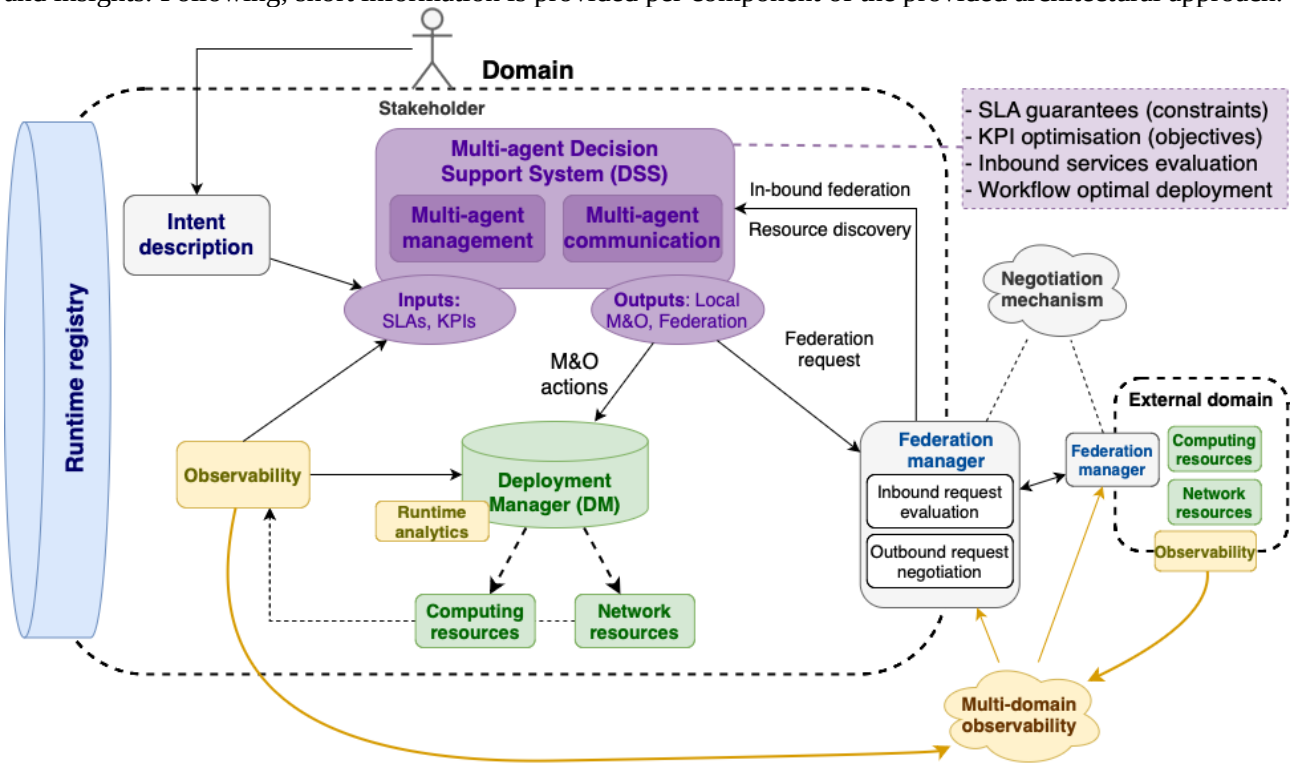

Figure 2-3: Multi-agent decision-making and interactions.

The **Runtime Registry** acts as a centralised repository that maintains updated information about the system's resources, services, and configurations. It facilitates resource discovery and provides inputs for managing and optimizing operations within the multi-agent system. The **Intent description** component captures the stakeholder's high-level requirements, goals, and objectives, such as SLA constraints or KPI targets. It serves as an input to the Multi-Agent Decision Support System (DSS), enabling the system to translate intents into actionable decisions.

The **Multi-agent Decision Support System** (DSS) takes up to manage the deployment and communication needs of the agents that govern all available resources within a domain to guarantee for:

- *Satisfying the SLAs of the registered services.*
- *Optimizing domain KPIs*: domain specific metrics, e.g., average power consumption, CPU utilisation, etc.
- *Evaluating inbound service federation*: if incoming (from other domains) services are accepted for deployment.
- *Deploy and maintain services across domains*: Decide where to deploy workflows either by local deployment or federation to external domains.

The DSS takes as input i) the specifications provided by the domain stakeholder in the form of an intent and ii) the SLAs of the deployed services (from this or other domains), and generates decisions regarding the deployment and lifecycle management of the services. Thus, its inputs and outputs are as follows.
Inputs:

- Domain internal workflows for deployment and the corresponding SLAs.

- Domain KPIs as objectives and constraints.
- Workflows/Services to be deployed from external domains and their SLAs.

Outputs:

- Service deployment.
- Scaling, migration, offloading, and/or load balancing directives.
- Service rejection (e.g., in case of insufficient resources).

The internal structure of the DSS based on the multi-agent approach revolves around the set of agents that tackle the corresponding orchestration problems. The deployment, operation and maintenance of the agents requires a dedicated **agent management** mechanism taking into consideration issues like timing/synchronisation, storage/ML models management, agent mapping to application and infrastructure, etc. Multi-agent systems also develop heavy communication requirements for synchronizing, message passing and collaborating towards common objectives, so a specifically designed **agent communication** mechanism is necessary for automating the cooperation of agents in the domain.

An implementation of the agent management and communication mechanisms should include the following characteristics:

- Management
  - *Deployment:* A way to deploy the agent as an operator managing specific resources.
  - *Observability interface:* A well-defined interface providing accessibility to the needed observations for decision-making.
  - *Orchestration interface:* A well-defined interface for orchestrating at runtime the resources that the agent is responsible for.
- Communication
  - *Shared information:* A common dataspace for agents to share observations and objectives to facilitate collaboration.
  - *Messaging:* A messaging mechanism to enable message passing between agents based on the deployed algorithmic approaches (e.g. Multi-Agent Reinforcement Learning).
  - *Synchronisation:* A protocol for synchronisation of agents in the decision-making process.
  - *Collaboration:* A way to define and enforce synergies between the agents on different levels, i.e. centralised, hierarchical, peer-to-peer, etc.

The **Deployment Manager** acts as the multi-cluster / network infrastructure manager responsible for managing the deployed services at runtime and resolving functional issues for guaranteeing the services' smooth operation. Its main objective is to implement the decisions made by the DSS and manage the computing and network resources available in the domain. The **Federation Manager** is responsible for offloading services/workflows across other domains and for making sure the corresponding SLAs are satisfied or if there is a need to re-negotiate the federation, starting a new decision-making round by DSS. It plays two main roles, provisioning the offloaded services in other domains and negotiating the terms of inter-domain federation. The negotiation process works in two directions, negotiating the deployment of services to other domains and negotiating the federation of external services to the domain's resources.

A **Multi-domain observability** stack is responsible for providing inter-domain observability of the services and the resources to the dedicated monitoring services. Although local observability servers may be at play, the need to monitor resources in other domains is apparent when service federation is enabled. Thus, a distributed observability system is required to allow this exchange between different players of different authorisation and access to the observed datasets. **Runtime analytics** are crucial when it comes to the continuous and undisturbed operation of the deployed workflows and services. Based on the observability stack, the troubleshooting modules are part of the runtime environment of the Deployment Manager responsible for guaranteeing the normal operation of the services.

2.1.1.2 Decentralised orchestration

The main objective of the decentralised M&O approach is to provide an overall solution to manage and orchestrate network services and resources through the whole network continuum, with a focus on integrating the vast number and diversity of devices in the extreme-edge domain, along with the numerous service components that may be deployed on them. This objective is considered to be impractical relying on a single

centralised system. E.g., just making central the collection and processing of monitoring data from such a wide range of resources would be highly challenging. Besides, as it is well-known, fully centralised approaches could lead to other issues, such as difficulties in the capacity planning for non-owned resources (the extreme-edge includes resources beyond the stakeholder own domain), the risk of a single point of failure, scaling issues, and the increased operational cost due to the complexity of managing a such large network. However, the decentralised M&O approach described here can optimise resources more effectively: unlike the common stakeholder-centric solutions that apply a pre-defined common set of M&O mechanisms and resources to all possible services in a uniform way, the decentralised method adapts M&O mechanisms to each specific service, which may have different needs based on their functionality and scope. Additionally, it is inherently "multi-domain," allowing service chaining across multiple domains using exposed interfaces in a cloud-native way relying on the microservices federation concept [GKV+19] [FSP+20]. This contrasts with centralised systems, which would depend on complex business and technological agreements between stakeholders to connect their M&O frameworks.

As anticipated in Deliverable D6.3 [HEX224-D63] the decentralised orchestration approach involves deploying multiple M&O systems throughout the network continuum. Just one of these systems, called the Common Infrastructure Management and Services Provisioning System (CIM&SPS), is shared, while others are customised for each network service to provide specific service assurance mechanisms. As explained in [HEX224-D63], the key idea is to separate the service onboarding process (a common feature of the network) from the services assurance, which can be handled individually for each service with a more lightweight and decentralised approach.

The CIM&SPS is considered made up of four so-called distributed network stakeholder support services, already introduced in [HEX224-D33] and [HEX223-D32]. They are the following:

- The **Infrastructure Registry Service** (IRS), which would act as a distributed database for resource orchestration, maintaining an accurate, real-time registry of network resources, despite their heterogeneity and volatility. In addition to the registry, it would also include automatic infrastructure discovery mechanisms to explore and store updated information about available virtual or physical resources in the network, considering the different device types, processing and storage capacity, reliability, or physical location, among other relevant parameters.
- The **Infrastructure Status Prediction Service** (ISPS), which would enhance the information of the IRS by providing insights into the future availability of infrastructure resources, enabling proactive M&O mechanisms. This ISPS would predict whether a specific infrastructure node, which could host certain service components, will remain available or maintain the necessary resources over time. Its operation would be based on data analytics, which may include AI/ML algorithms, to generate future availability information based on the current state of infrastructure resources.
- The **Deployment Service** (DS), which would be responsible for deploying network services made through the aggregation of microservices. The deployment descriptor of each of these microservices would outline the requirements for the infrastructure devices on which they should be deployed, in such a way that, relying on the information from the IRS and the ISPS, the DS would identify the best match between these requirements and the available infrastructure resources to optimise deployment. This DS could aggregate microservices from multiple parties to implement the microservices federation concept mentioned above [HEX224-D33].
- The **Services Registry Service** (SRS), which is a support service for the DS and also functions as a registry through a distributed database, similar to the IRS. However, the SRS specifically tracks the current execution environment of the already deployed services, considering the high volatility of the extreme-edge domain, by providing real-time information on the locations of network service components, even if they migrate between infrastructure nodes. The SRS is considered ancillary because some network services may be designed to deploy entirely or partially on stable non-volatile nodes, making the use of the SRS unnecessary in those cases.

These four stakeholder support services enable the provisioning of new network services and are respectively implemented relying on four specific components, namely: Deployment Nodes (DN), Infrastructure Registry Nodes (IRN), Service Registry Nodes (SRN), and Infrastructure Status Prediction Nodes (ISPN). Multiple instances of these components would be distributed across the whole network to make up each support service (e.g., multiple instances of DNs or IRNs would respectively form the above-mentioned DS and IRS), as

depicted in Figure 2-4. These nodes would be hosted by different stakeholders part of the continuum with the capabilities to house and manage these network elements (e.g., MNOs, vertical industries, hyper-scalers, or others).

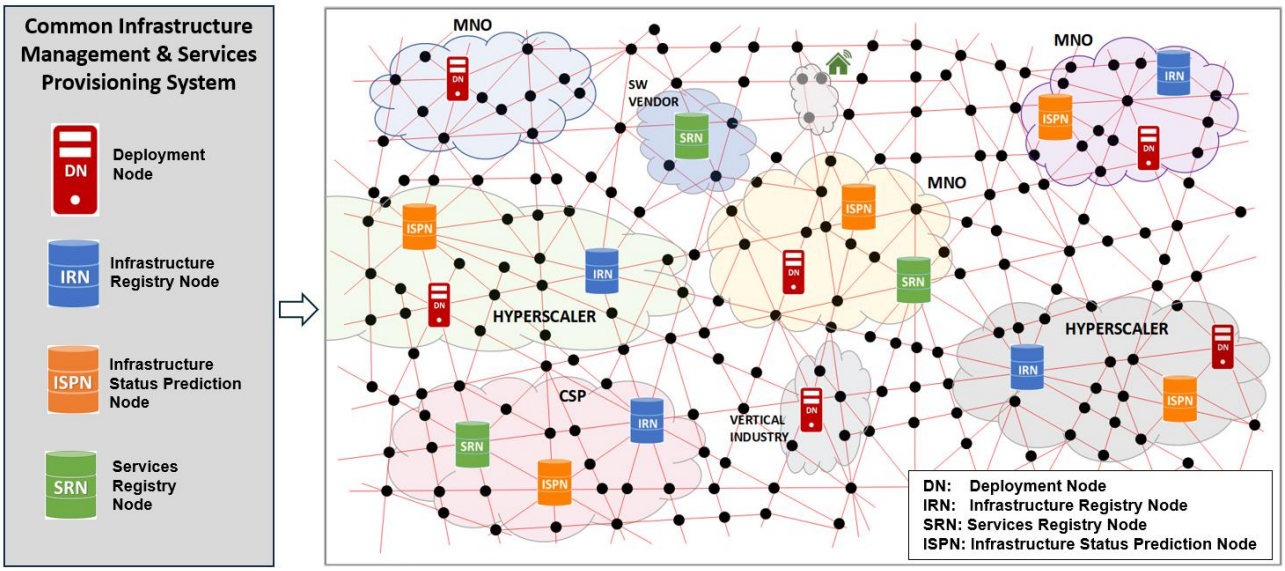

Figure 2-4: Common Infrastructure Management and Services Provisioning System [HEX224-D63].

Beyond the CIM&SPS, and regarding the tailor-made service-specific assurance mechanisms also mentioned above, the decentralised orchestration approach allows for flexibility in implementing service-specific assurance mechanisms. It does not dictate a specific method for these mechanisms, enabling service designers and developers to design them based on the unique needs of the network services they must support. While multiple network services may adopt a common set of services assurance mechanisms, the primary focus is to empower developers to select the most suitable approach tailored to the distinct requirements of each service, which could vary significantly.

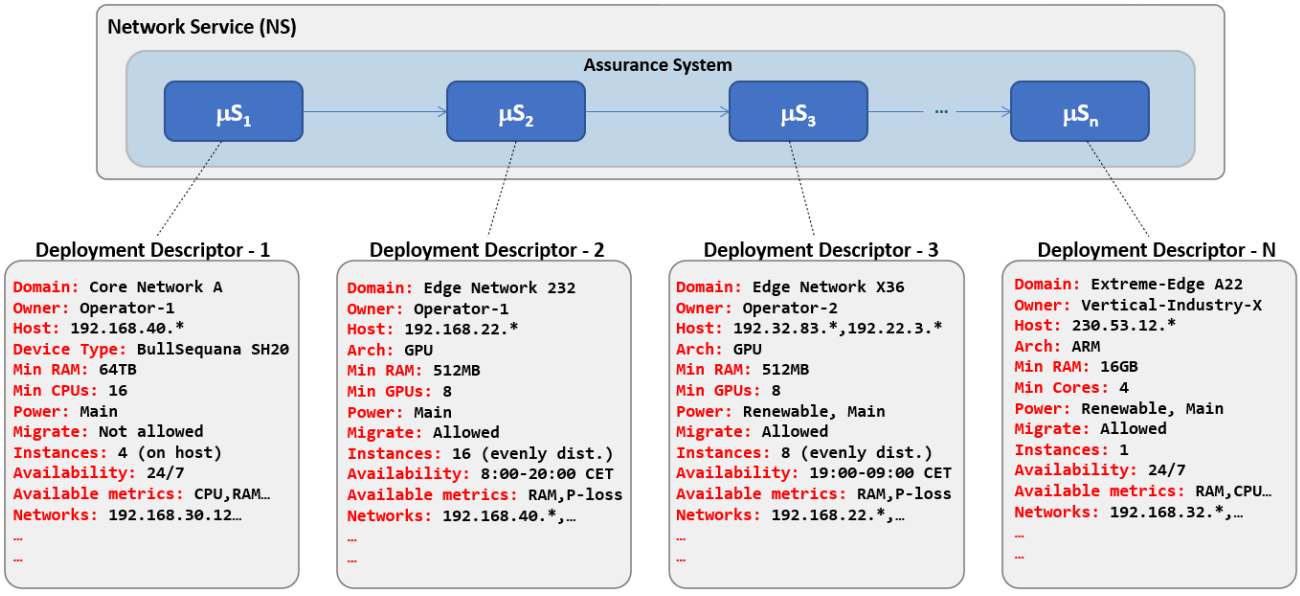

Figure 2-5: Network Services definition for the decentralised M&O approach.

As introduced in [HEX224-D63], those mechanisms will be integrated into network services themselves, ensuring that specific functionalities are in place to keep the required availability of service components, facilitating their migration (if necessary), the collection of metrics, and providing access to the involved stakeholders. This integration would allow for a high flexibility and scalability, with the M&O assurance mechanisms distributed throughout the network alongside the services they support. Various state-of-the-art technologies can be utilised to implement these per-service M&O mechanisms, including container

management solutions (e.g., [K8S] [K3S] [NOMAD] [SWARM]), custom-developed systems, microservices choreographies (i.e., self-orchestrated microservices not relying on any external orchestration component) [CDT18], or other M&O solutions that could be available in the future.

Figure 2-5 illustrates the design approach for defining and deploying network services composed of various microservices, along with the embedded tailor-made M&O mechanisms (Assurance System block). The figure also shows the deployment descriptors of each microservice, suggesting a potential data model for specifying deployment requirements, such as the need for devices in specific network domains, computing or storage capabilities, the number of instances to be deployed, power supply types, etc.

2.1.2 Overall Functionalities

This section describes five overall functionalities designed to address the M&O needs across the entire network continuum. They are the following:

a) The Management Capabilities Exposure (MCE) functionality, which serves as a specialised API connector, enabling secure and scalable interactions across M&O systems with seamless, event-driven communication, in line with the ETSI ZSM Integration Fabric specification [ZSM-002].
b) The Real-Time Zero-Touch Control Loops Automation and Coordination functionality. This functionality is aligned with the ETSI ZSM 009 specification ([ZSM-009-1], [ZSM-009-2], [ZSM-009-3]) and focuses on automating network management through closed-loop control. These control loops are implemented as general-purpose configurable software artifacts, allowing them to adapt to various network, infrastructure, or service-level objectives. The automation system can also connect with AI/ML algorithms to enable predictive actions, making network management more autonomous and responsive to real-time changes.
c) The Monitoring and Telemetry functionality, which supports a wide range of monitoring protocols and enables the integration of user-defined metrics across various network domains. This functionality can also feed multi-domain, heterogeneous data into AI/ML models, enabling advanced decision-making processes. By correlating large volumes of data from different sources, this functionality can enhance proactive M&O operations, supporting the automation of network management tasks.
d) The SLA-driven Federated Orchestration functionality provides dynamic SLA creation and policing mechanisms using Distributed Ledger Technologies (DLT) -based smart contracts to facilitate service continuity beyond a service provider network.
e) The Trust Management system, intended to incorporate trust management functionalities directly into the orchestration processes. This involves methods to measure and evaluate trust levels, which are factored into workload allocation decisions. By incorporating trust metrics, the system ensures that resource distribution decisions are both secure and efficient, particularly within the multi-stakeholder environment that characterises the network continuum.

The following subsections describe in more detail these functionalities.

2.1.2.1 Management capabilities exposure

The Management Capabilities Exposure (MCE) functionality is an important component of the Smart Management Framework. It functions as a crucial connector, establishing communication channels between API producers and consumers, and providing a single-entry point for external communications. This ensures that communications are directed to the appropriate components within the framework, thereby ensuring seamless connectivity and reachability.

Designed to be in line with the cloud-native and the plug-and-play concepts, the functionality is easily configurable and adaptable to various network scenarios. It supports asynchronous and event-driven interactions within and outside the management framework scope, allowing for the dynamic addition of new services and the real-time exposure of specific capabilities. The overall system components and system architecture were already detailed in D6.3 [HEX224-D63].

The MCE concept is inspired by the ETSI Zero-touch Network and Service Management (ZSM) specification [ZSM-002], and particularly, the Integration Fabric concept. This functionality facilitates comprehensive integration and interoperability between diverse M&O entities within a network environment. Acting as a

centralised middleware layer, it supports efficient communication and data exchange, fulfilling operational goals such as connection, dependability, security, and observability. The dynamic registration and discovery of network elements and services are integral to this functionality, adhering to the plug-and-play concept. This capability streamlines operational aspects by automating the registration and discovery process, thereby maximizing resource utilisation and optimizing operating expenses. These features are crucial for improving overall operational performance and ensuring seamless integration across various administrative and operational domains. The cloud-native principles, event-driven architecture, and modern *APIfication* practices ensure comprehensive integration and operational efficiency, aligning with the design principles of service separation, reusability, and cloud compatibility, as outlined in ETSI ZSM specifications.

One of its key aspects is the support for such *APIfication*, transitioning from traditional, tightly coupled integrations and legacy protocols to loosely-coupled APIs. This transition is facilitated by adhering to OpenAPI 3.0 standard[4], which provide clear and manageable interfaces for scalable, containerised microservices. The modular and stateless nature of the MCE system allows for deployment as scalable, containerised microservices, enhancing its adaptability and efficiency. This dynamic and flexible management structure enables seamless communication and integration across various administrative and operational domains, optimizing resource utilisation and minimizing operational expenses.

The realisation of the communication goals of this functionality, instead, are reached thanks an event streaming platform, serving as a message broker that enables the event-driven communication model. As a shared message bus, it allows the framework's internal components to interact seamlessly and orchestrate the operations of various parties participating in the communication. Data streams within the framework are managed topic-oriented, utilizing the concept of queues. A queue is a logical stream of records belonging to a topic. Usually, in an event streaming platform, the queues are divided into partitions for efficient data management. This partitioning allows data to be replicated over multiple brokers and enables multiple consumers to read data simultaneously, providing parallelism, horizontal scaling, and fault tolerance. By supporting large volumes of data with high throughput and low latency, this design ensures data redundancy, fault tolerance, and strong data durability through flexible retention policies.

On the other hand, security is also supported by the MCE. The MCE ensures authenticated and authorised access to network resources, managing tokens and keys for various quality-of-service levels. Secure communication is maintained through transport layer security (TLS), preventing information leakage and safeguarding data integrity. The inclusion of a security module guarantees that all interactions are secure and compliant with stringent industry standards.

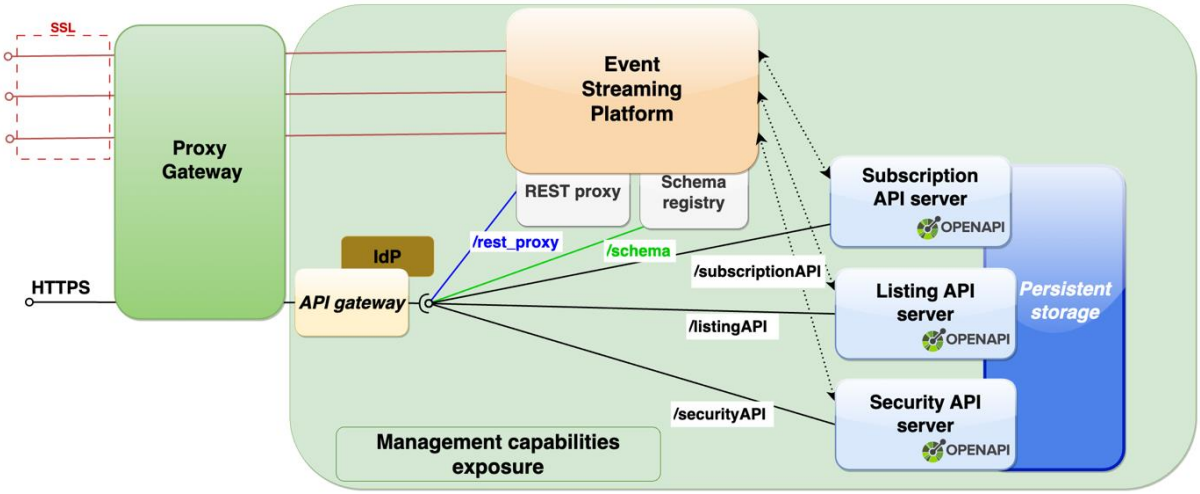

Figure 2-6: MCE internal architecture and interfaces

[4] OpenAPI 3.0 is a widely adopted specification for describing RESTful APIs. It provides a standardized format for documenting API endpoints, request/response structures, and authentication methods, enabling better interoperability and easier API integration for developers.

Figure 2-6 shows an overview of the MCE structure. More details about the API and the functioning, supported by dedicated workflows, were already provided in D6.3 [HEX224-D63].

2.1.2.2 Real-Time zero-touch control loops automation & coordination

Real-time (RT) zero-touch control closed loops (CL) constitute the foundation of network automation, providing the building blocks to bring advanced intelligence in self-configuration, self-adaptation, and self-optimisation capabilities of future mobile networks. The usage of CLs is crucial to introduce autonomous control and orchestration in networks with multiple technologies and high complexity. Following the CL paradigm, concurrent multi-objective tasks can operate the network to jointly optimise resource utilisation, energy consumption, and service performance in a scalable and efficient manner. CL functions can be specialised for different scopes and work at different layers or domains. Their implementation makes use of pervasive data analysis and AI/ML, exploiting network programmability for dynamic re-configurations able to reach an extreme level of granularity. The interworking with complementary enablers and technologies (e.g., Digital Twins or Intent Management) can enhance CL performance and capabilities, facilitating autonomous adaptation to several categories of intents and early identification of potential conflicts between coexisting CLs, with the possibility to test alternative strategies for their resolution or mitigation in realistic and accurate sandboxes.

CL modelling and management in Hexa-X-II derive from and extend the current work in ETSI ZSM ISG [ZSM-009-1][ZSM-009-2][ZSM-009-3]. The project has developed concrete examples of CLs applied to different domains and objectives, as well as practical implementations of CL Governance and CL Coordination functions for management of CL components, as documented in D6.3 [HEX224-D63].

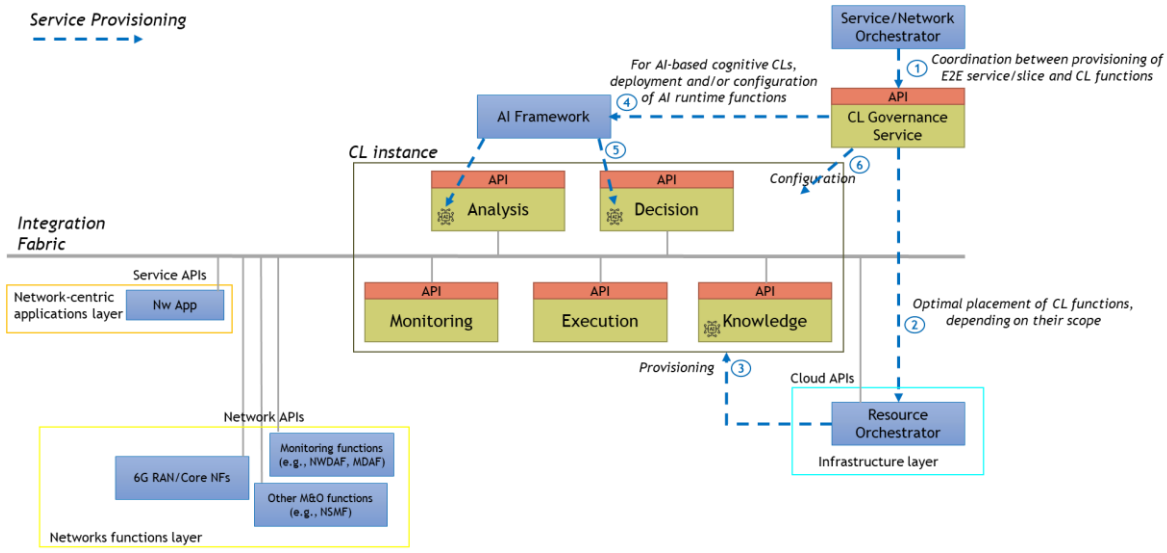

Figure 2-7: CL functions and CL Governance during CL provisioning.

CLs are decomposed in several cloud-native management functions, in compliance with the Service-based Management Architecture (SBMA) [28.831] approach. These functions are combined to deliver the four stages of a generalised CL, i.e., Monitoring, Analysis, Decision, and Execution, with the complementary so-called Knowledge element for storage and sharing of CL-related data among CL internal functions. The interfacing between CL functions can be mediated through the MCE functionality (Sec. 2.1.2.1) through an Integration Fabric or a data bus in general. In this Hexa-X-II approach, CL functions are handled within the M&O procedures and they are dynamically provisioned and orchestrated over the continuum infrastructure by the CL Governance function. CLs can be instantiated as part of the virtual entities they help to manage, e.g., at slice or service level, and their provisioning can be triggered from service orchestrators or network slice management functions that interact with the CL Governance, as shown in Figure 2-7. It should be noted that, for AI-driven CLs, the CL Governance may interact with the AI framework to request the deployment of AI/ML functions or the loading of trained ML models to be used in the analysis or decision stage. CL functions

follow a common pattern, with unified interfaces and modelling, to enable interoperability and facilitate their deployment and composition in multi-vendor scenarios.

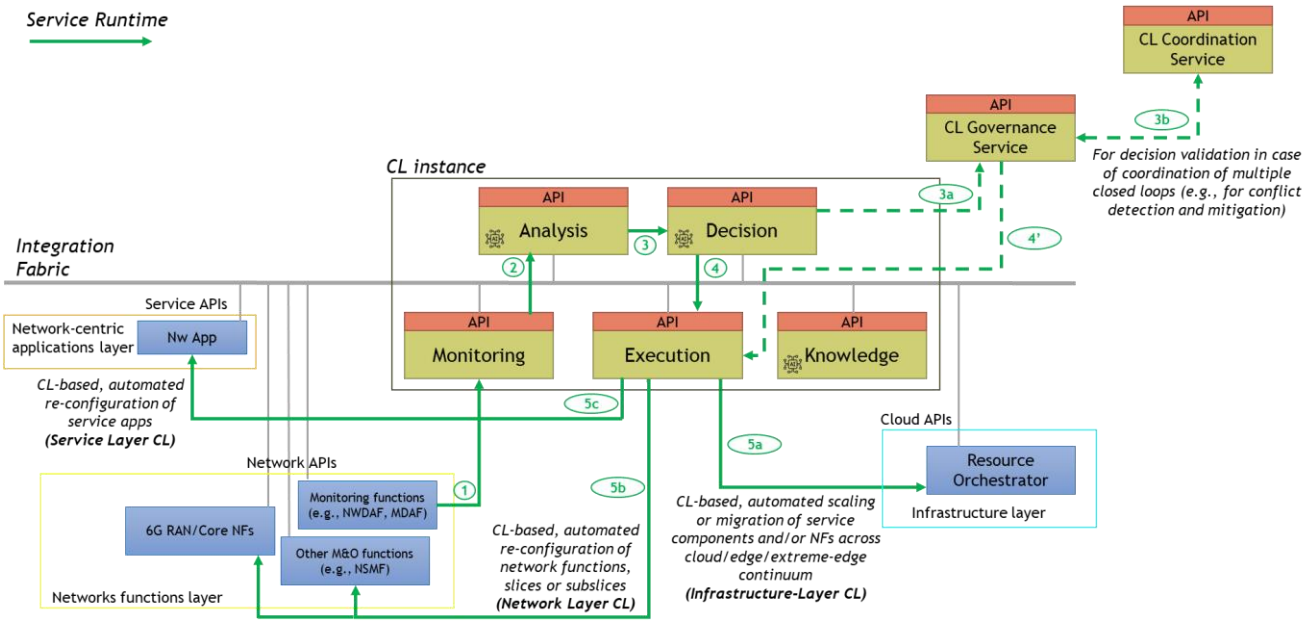

Figure 2-8: CL functions and CL Governance during CL runtime.

At runtime (Figure 2-8), target monitoring data are retrieved from various sources, including network analytics functions, computing platforms, or network applications. These metrics, collected and pre-processed at the CL monitoring, feed the CL analysis to extract statistics, insights, predictions, etc., which are then evaluated at the CL decision stage. Here, the CL takes decisions on potential re-configuration or lifecycle actions to be applied on the managed entities. The decision may immediately trigger the CL execution stage, or may require a preliminary validation, e.g., for multi-CL coordination purposes. In this case, the decision is notified to the CL Governance mechanism, which in turn interacts with the CL Coordination function to take a final decision on the proposed actions. The detailed workflows and information models are documented in D6.3 [HEX224-D63].

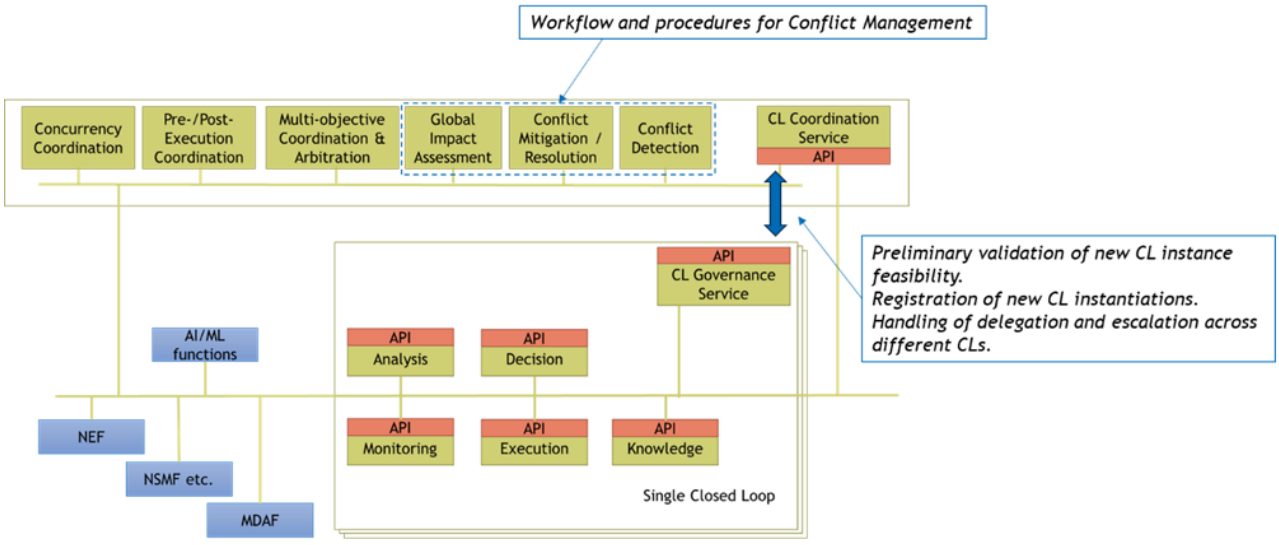

Figure 2-9: CL coordination functions.

Considering the level of specialisation associated to a given CL, it is crucial to enable an efficient and scalable execution of multiple CLs working in parallel, each of them responsible for a given objective, but in most cases presenting interdependencies among them. Decisions taken from a CL may have a direct or indirect impact on managed entities controlled by other CLs, leading to temporary unstable situations or, even worse, to conflicting configurations. In this context, coordination strategies must be applied to guarantee the overall

consistency of commands from multiple CLs. For example, the execution of CL actions may need to be ordered, aggregated, or synchronised together; conflicting decisions may need to be validated and prioritised through an arbitration schema; the objectives of multiple CLs may be combined to reach global decisions, etc. The cooperation among multiple CL instances is handled through the CL Coordination mechanisms, which can be implemented with several M&O functions working together. Examples are shown in Figure 2-9, with functions for conflict management, assessment of global impact of more CLs, combination of CL objectives, and coordination of CL actions.

Two coordination models are considered: peer-to-peer and hierarchical interactions. The former is suitable for CLs working at the same level, where CL coordination is in charge of ordering and scheduling concurrent execution actions or validating the combining effect of multiple decisions to reduce potential conflicts. The latter model is adopted in scenarios where there are vertical dependencies between the CLs, usually with a parent CL that can delegate specific tasks to one or more child CLs working at a lower layer. In this case, the CL coordination is in charge of handling delegation commands and notifications of escalation requests.

2.1.2.3 Monitoring and telemetry

Telemetry refers to the process of automatically collecting and transmitting data from remote systems or devices, while monitoring is the activity of observing, analysing, and interpreting that data. Programmable network monitoring and telemetry, powered by Software-Defined Networking (SDN) and automation technologies, revolutionise network data collection and analysis by ensuring continuous oversight and automated real-time data transmission from diverse sources [ETS20]. This functionality builds on the Network Data Analytics Function (NWDAF) defined in 5G [29.520], as it extends capabilities for 6G networks, incorporating multi-source monitoring data and addressing immediate event reception, forwarding, fusion, and processing, with a scalable cloud-scale architecture. The functionality, featuring multi-layered data collection from virtual and physical components and applications, enhances decision-making by providing comprehensive operational insights and uncovering network utilisation patterns for optimised configurations and cost-efficiency. It also measures energy consumption to develop energy-efficient algorithms. Leveraging various technologies and protocols, it also incorporates sequential processing for efficient metrics and alerts handling, ensuring smooth and scalable operations.

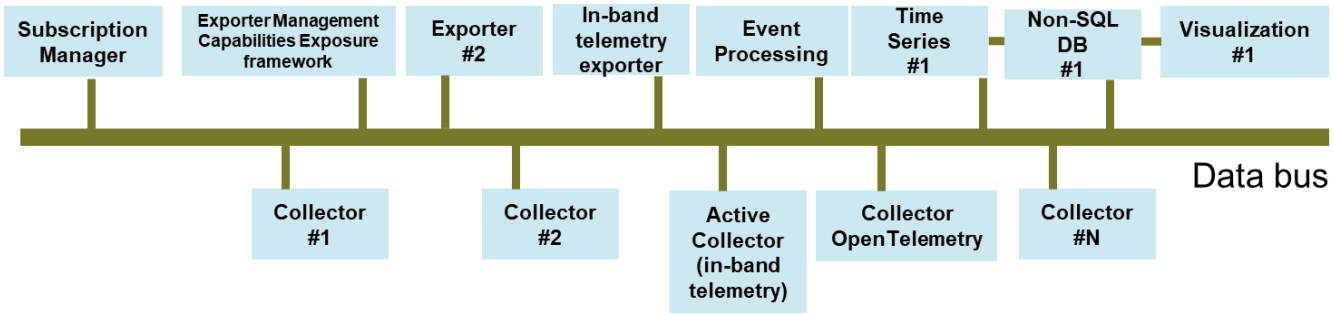

Figure 2-10: Monitoring and telemetry functionality architecture.

As a whole, the Monitoring and Telemetry functionality architecture consists of a cloud-native design with microservices connected through a common data bus. Figure 2-10 depicts a schematic of data processing system, arranged horizontally to illustrate the flow and interaction of data among various components (more details were provided in [HEX224-D63]). At the forefront of the system is the so-called Subscription Manager, which oversees the configurations for data collection across the system. It can also apply the required security and access control mechanisms. Next in line are the Exporters, which denote distinct pathways through which data is extracted and possibly disseminated to different destinations or for various uses. For example, an exporter to the MCE functionality is considered. An In-band telemetry exporter is also featured, implying a mechanism tailored to handle telemetry data within the operational bandwidth, ensuring efficient monitoring.

As we move further right, there is also an Event Processing unit, indicative of a subsystem dedicated to handling discrete events that occur within the system — these could be irregular, significant, or require special processing separate from the main data flow. Adjacent to this is a Time Series database, which provides storage

and retrieval functionality for data points collected sequentially over time, a common requirement for monitoring trends and patterns.

The Non-SQL database (DB) component indicates the use of a non-relational database, which is designed to store and manage large volumes of unstructured or semi-structured data, signifying flexibility and scalability in data handling. Finally, the Visualisation module provides an endpoint for the system's data, where it is likely transformed into graphical representations to aid users in interpreting and analysing the data.

Supporting these main components are several "Collector" blocks. These are depicted beneath the main data pathway, suggesting a hierarchical relationship where these collectors feed into the upper-level components. Among these, an Active Collector (in-band telemetry) is specified, highlighting its active role in gathering in-band telemetry data, which is crucial for real-time monitoring and analysis. Also, other specific state-of-the-art collectors (e.g., OpenTelemetry-based collectors [OTL24], more information provided in Annex A.2 Open Telemetry and Data Fusion) can be considered, to allow instrumentation, generation, collection, and export of telemetry data (metrics, logs, and traces).

Several specific implementations and validation of this functionality are considered in Section 2.2.1.7 Monitoring and Telemetry Implementation.

2.1.2.4 SLA-driven Federated Orchestration

In order to provide service continuity in areas where the consumer domain does not have sufficient coverage or is completely absent, roaming agreements are commonly established among service providers based on agreed service level requirements. With the advent of microservices and the cloudification of provider networks such roaming scenarios become more complex, requiring agile mechanisms between peers so that services are not interrupted. Furthermore, relying on closed loop network control (Sec. 2.1.2.2) requires the automation of most aspects of the service life cycle. In this scope, this SLA-driven Federated Orchestration functionality would work to dynamically establish roaming agreements between providers in this new cloud-native paradigm without a human in the loop. It therefore addresses both challenges posed by this paradigm shift.

The main idea behind this functionality involves employing a permissioned blockchain [ABC23] where each administrative domain operates a single node. A unified so-called federation Smart Contract (SC) would be deployed on this blockchain to serve as a decentralised authority, ensuring transparency and trust. Each node runs the same instance of the federation SC, guaranteeing coordinated code execution and bolstering security and trust.

The federation SC is essential for protecting sensitive information while managing federation procedures. To join the network, a new domain must register with the federation SC, providing its unique blockchain address, administrative details, and service footprint. Domains interact with the federation SC via a specific API, enabling to participate in the federation processes as both, consumers or providers.

As shown in Figure 2-11, upon a consumer domain announcement or federation offer, the federation SC would log it as a new auction and would broadcasts it to all registered domains. The announcement would include the desired QoS of the service, e.g., maximum allowed latency, minimum service availability, etc. To protect privacy, the consumer domain's address would be concealed, using a single-blinded reverse auction mechanism where consumer domains would anonymously create offer announcements and provider domains would place bids.

The federation SC would oversee the process, allowing only the consumer domain to close the bidding, thereby empowering the consumer to apply their selection criteria. Bid offers would include the pricing information as well as the QoS that the provider could guarantee. The consumer domain would periodically check the federation SC for bids, select a provider, and close the auction. The federation SC would also record the chosen provider as the winner, notify that, and announce the auction's completion.

Subsequent direct communication and information sharing would occur between the consumer and selected provider domains. The federated service would be deployed and integrated into the end-to-end service by the consumer domain. Upon completion, the provider domain would initiate the billing for the service. The same permissioned blockchain network could be used, incorporating micropayment channels to ensure unbiased and immutable billing records for both, consumer and provider domains.

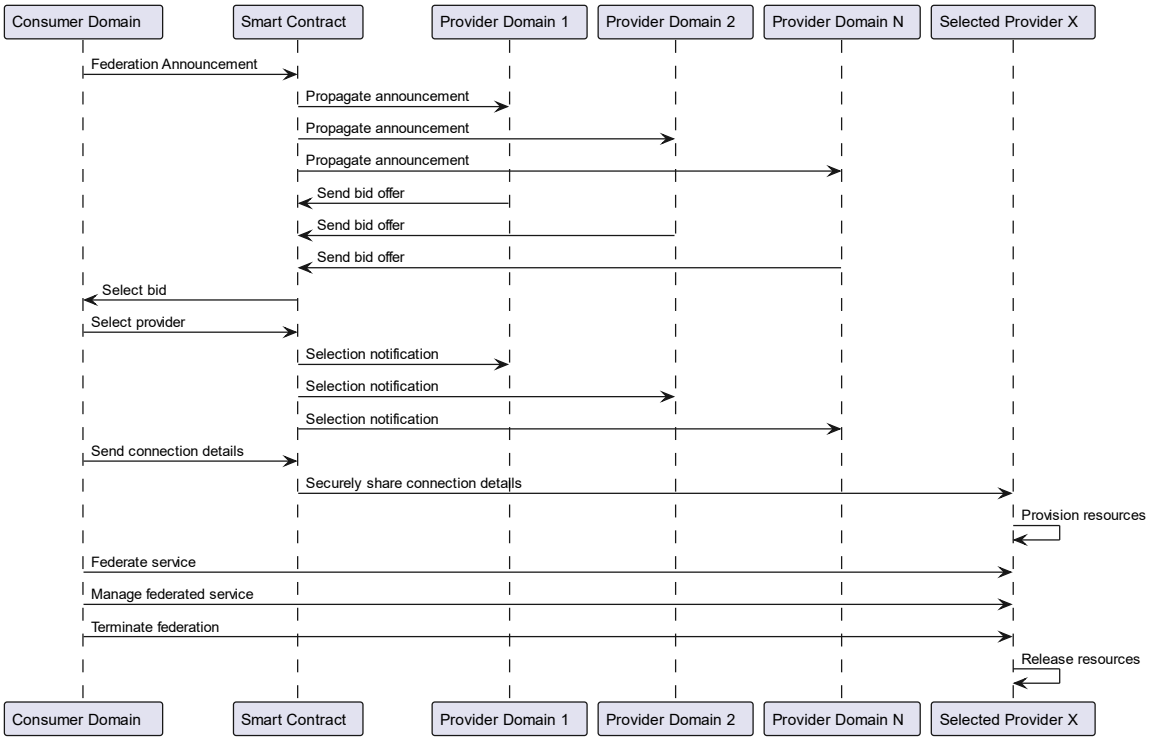

Figure 2-11: Sequence diagram for SLA-driven orchestration.

2.1.2.5 Trust Management

6G networks are anticipated to provide extensive connectivity, exceptional reliability, and high mobility to IoT devices at the extreme-edge domain. In these environments, numerous diverse devices constantly exchange information, often over unstable networks. To address this, a flexible, lightweight, and adaptive access control mechanism is considered needed to ensure secure communications among trusted devices.

The Trust Management System functionality introduces *Trust Evaluation Functions (TEFs)* to assess the trustworthiness of infrastructure components, compute nodes, services, applications, and third-party consumers. It also introduces the *Level of Trust Assessment Function* (*LoTAF*) to evaluate the trustworthiness, or *Level of Trust* (*LoT*) of network services in a particular application environment. The Trust Management System functionality combines the outcomes from the two functions, TEF and LoTAF, producing a unique trustworthiness index per entity (e.g., edge node, router) of interest.

TEFs can be used to estimate the trustworthiness of the system based on specific use cases. For workload placement and orchestration, a TEF focusing on the infrastructure layer evaluates the trustworthiness of compute nodes. For flexible topologies, a TEF for flexible networks and nodes is used. For service-centric orchestration, a TEF targeting the service layer is applied.

Infrastructure layer TEF can be used by cloud orchestration engines to allocate workloads to compute nodes, aiming for maximum trustworthiness. It receives input from monitoring telemetry component, it analyses the data related to compute nodes and computational workloads placed on them, and it produces trust indexes for these compute nodes using a novel mathematical formula described in [HEX224-D63]. This output is then fed to orchestration components (e.g., functionality allocation), which, along with other metrics and data, can suggest the optimal placement of computational workloads to available compute nodes.

Each trust index produced by the TEF for the infrastructure layer is the weighted sum of the:

- Availability: Percentage of the time that a compute node is not fully loaded or unavailable.
- Reliability: The success rate of the compute node in executing tasks within a time threshold.
- Security: Related to secure communication and trusted computing, encoded as a binary variable indicating the availability of Transport Layer Security (TLS).
- Multi-connectivity capabilities: A range from 0 to 1 based on the supported technologies (e.g., 4G/5G, Wi-Fi, NB-IoT).

- Battery level: Status of battery-powered devices/nodes (e.g., robotic units in industrial environments).

The Trust Management System functionality also proposes the *Level of Trust Assessment Function* (*LoTAF*) to evaluate the trustworthiness, which is used to assess the *Level of Trust* (*LoT*) [HEX224-D63] of network services in a particular application environment. Overall, LoTAF intends to facilitate the management of electing distributed resources and infrastructures in the network continuum. In particular, LoTAF is a neutral and intelligent service to check how much confidence may be attributed to network services before they are instantiated and used as well as assessing the initial LoT whereas an end-to-end connection is active. Therefore, it aims at ensuring trustworthy E2E connections across multiple network domains and providers. It is worth mentioning that LoTAF is not a one-time process, but a continuous process that runs during the whole life cycle of a trust relationship, so it continuously collects trust indicators to increase or decrease the LoT based on reward and punishment mechanisms.

When it comes to the LoTAF life cycle, it pursues two well-known standards. On the one hand, the ITU-T Y.3057 [ITU-Y.3057] supports the management of trust systems ICT infrastructure/services. In addition, it also introduces the principal phases of LoTAF: i) trust specification, ii) trust analytics, iii) trust establishment, iv) trust update, and v) trust termination (see detailed information in Annex A.1). On the other hand, the Service Assurance for Intent-Based Networking (SAIN) architecture [SAIN-23] contributes to monitor trust requirements and ensure an appropriate health status of network services. SAIN is also the engine that drives LoTAF to obtain quantitative attributes as well as guarantees the integrity of services through symptoms (system information and telemetry) settled in the Trust Level Agreement (TLA). By means of continuously monitoring trust indicators, LoTAF may support other systems, for example, the user-centric service provisioning systems, since it may enhance the SLA enforcement via its continuous monitoring process, which guarantees service assurance in terms of trustworthiness. Likewise, LoTAF may also support the decentralised orchestration system. In this case, one of the LoTAF phases, trust analytics, may help network services orchestration on the network continuum, as it enables an initial assessment of available services in a catalogue considering user intent requests with respect to trust. In this vein, LoTAF may carry out an initial pre-filtering of available network services (together with their owners) and rank them based on the initial Level of Trust. Furthermore, it may also check the network service assurance once the selected candidates are up and running.

Concerning trust indicators, LoTAF conducts evaluations of both objective trust indicators (quantifiable aspects) derived from the RFC 9417 [SAIN-23] theoretical model, and subjective trust indicators (qualitative aspects), which could be obtained from recommendations by third parties, personal judgments, or reputation. These elements are initially gathered through a monitoring agent during the first stages of assessing trustworthiness, i.e., the analysis of a set of available network services in a catalogue. In subsequent stages, when a trust relationship would be established, only the quantitative parameters would be used to consistently revise the initial LoT determined earlier. After collecting all trust indicators, LoTAF leverages Bayesian processing to compute membership degrees between LoT and affinity, with pre-defined thresholds established in accordance with the terms of the TLA. Then, LoTAF would release the final LoT, combining objective and subjective features, ranked between 0 and 1 (the closer to 1, the higher the confidence level assigned). Last but not least, LoTAF would share its trust index, or LoT, with the Trust Evaluation Function (TEF), which would measure a different set of features related to trustworthiness. Through this action, which would be bidirectional, the Trust Management System functionality may provide a unique trust index combining the outcomes from the two trust-oriented functions: TEF and LoTAF. Section 2.2.2.3 describes the implementation details of these concepts. It is worth mentioning that Annex A.1 also provides more detailed information about LoTAF beyond what was already reported in the previous D6.3.

To summarise, the LoT assessment function strives to guarantee security and trust properties in 6G network services, considering both TLA enforcement and TLA assurance. Similarly, the LoT Assessment Function can also intend to ameliorate both end-users and providers experience, and support automatic network adjustment and intelligent service provisioning.

Figure 2-12 shows the interactions within the Trust Management functionality, as well as the interactions with some of the related framework components. As the figure shows, it combines the output of TEF and LoTAF through the MCE functionality described in Section 2.1.2.1 to provide a unique trust index. Furthermore, TEF and LoTAF can also rely on data and metrics from the Monitoring and Telemetry functionality described in

Section 2.1.2.3 to get further information so as to compute a trust index. Such value is later leveraged by the Functionality allocation component to find out the optimal placement of computational workloads (more details in Section 2.1.4.2).

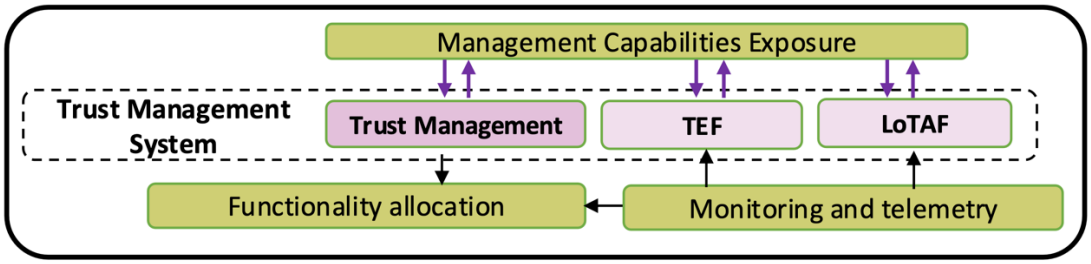

Figure 2-12: Trust management system architecture and interactions with other components in the management framework.

2.1.3 Specific Systems

This section describes the key systems operating at the stakeholder level in the management framework. Several important components are included here, namely:

- The 3rd party resource control separation system, designed to establish separate management spaces for each stakeholder in a multi-stakeholder environment, enabling precise control and operation over assigned services, applications, and resources while ensuring security, privacy, and trust.
- The User-Centric Service Provisioning system, which enables more dynamic and flexible SLA definitions.
- The network Digital Twins creation mechanisms. This system is intended to create virtual models of network environments, enabling safe pre-production testing of M&O systems and strategies at stakeholder level. These digital twins can also be connected to live production systems, making them interactive and capable of providing real-time insights.
- The Sustainable MLOps system, intended to automate, monitor, and optimise workflows for developing, deploying, and operating AI/ML-based network services, while monitoring the energy consumption within these workflows, in line with Hexa-X-II's sustainability goals.
- Network Programmability system, which integrates the SDN technology into the framework. Beyond the traditional benefits of SDN, it aligns with the cloud-native model and cloud continuum concept, while also providing new interfaces for emerging devices. This system is primarily based on the ETSI-hosted TeraFlowSDN project [TFS].
- The privacy protection for data analytics system, intended to ensure that sensitive data remains protected during AI/ML processes and decision-making, enabling privacy-preserving analytics.
- An AI/ML-based security control system for intent-based management systems, intended to support the intent-based management system addressed in WP2 [HEX224-D24].

In the following these specific systems are described in more detail.

2.1.3.1 Third-party resource control separation

The 3rd Party Resource Control Separation system is designed to create distinct management spaces per stakeholder within a multi-stakeholder environment. This enables precise operation and control over allocated services, applications, and resources while ensuring security, privacy, and trustworthiness towards 6G networks.

The system can be used to provision each stakeholder with a tailored management space that defines their permissions and controls their interactions based on predefined roles and profiles. It moves beyond static Role-Based Access Control (RBAC) and Lightweight Directory Access Protocol (LDAP) by introducing a model-

driven approach for more granular access control, which allows for detailed permission settings to avoid conflicts in resource-sharing environments.

Key components of the system include the resource owner, user, resource server, authentication function, and authorisation function. The resource owner grants access to protected resources, such as compute, memory, network, or services. The resource owner is often referred to as the Capability Operator (COP) [HEX223-D22]. The user is the stakeholder requesting access to these resources (Figure 2-14). The resource server hosts the protected resources and provides access through APIs, while the authentication function verifies identities, and the authorisation function determines what resources and actions a stakeholder is allowed to access.

The conceptual model of this framework (Figure 2-13) introduces access rules and permissions, extending traditional RBAC models. Access rules are granular sets of permissions specifying allowed or denied actions on protected resources. Roles are collections of these access rules, defining what resources a stakeholder can access and what actions they can perform. Identity represents stakeholder properties used for authentication and authorisation, associated with roles to define management spaces.

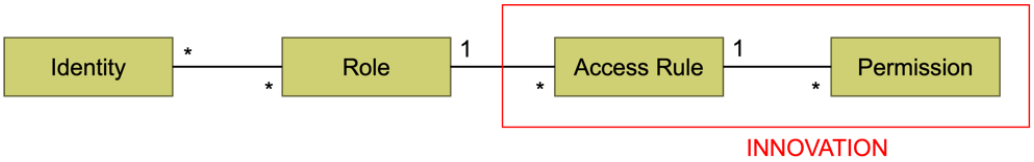

Figure 2-13: New information elements to extend RBAC model.

The Third-party Resource Control Separation system operates across three main stages: design time, onboarding, and operation time. During the design time stage, the system administrator (COP) creates granular access rules and associates them with roles representing different consumer types. These roles and rules would be stored in a Management Information Base (MIB). When a stakeholder registers, the onboarding stage would begin with the system using information from the "3P profiling"[5] component to create a tailored management space. This would involve mapping stakeholder information to an instance of the so-called "identity" class, associating it with predefined roles, storing it in the intra-stakeholder space, and generating a unique identifier for the stakeholder.

Once onboarded, stakeholders could access M&O capabilities via granular access control mechanisms. The access process could vary based on the protocol used. E.g., for REST/OpenAPIs, authentication and authorisation would be combined within a single server, using token-based techniques and optional OIDC (OpenID Connect) for authentication. Alternatively, in the case of Netconf/YANG, authentication would occur at the transport layer using mTLS (mutual TLS), and authorisation would rely on the Network Configuration Access Control Model (NACM).

The system considers intra-stakeholder spaces for individual management and inter-stakeholder spaces to manage policies and conflicts among stakeholders. The intra-stakeholder space would specify management spaces for each stakeholder, implementing the conceptual model with roles, identities, and access rules to ensure precise control over resource usage without conflicts. The inter-stakeholder space would manage policies to support multi-tenancy and address conflicts that may arise when multiple stakeholders are sharing resources. Policies would include priority-related policies, which tag management spaces with different priorities to resolve conflicts based on priority levels, and pre-emption policies, to enable pre-emption capabilities to address conflicts when priorities are the same, ensuring the system stability.

This system is considered a vital layer that separates external environments outside a stakeholder's domain (i.e., third-party stakeholders or untrusted systems requiring controlled separation), from internal resources, ensuring robust access control and conflict management in multi-stakeholder 6G network environments. By defining precise roles, identities, and access rules, it would enable secure and efficient resource sharing among stakeholders, aligning with the advanced requirements of future network systems (as shown in Figure 2-14).

[5] A process component used during onboarding to map stakeholder information to roles and identities, enabling tailored resource management in multi-stakeholder 6G systems. Refer to [HEX223-D22] [HEX224-D23]

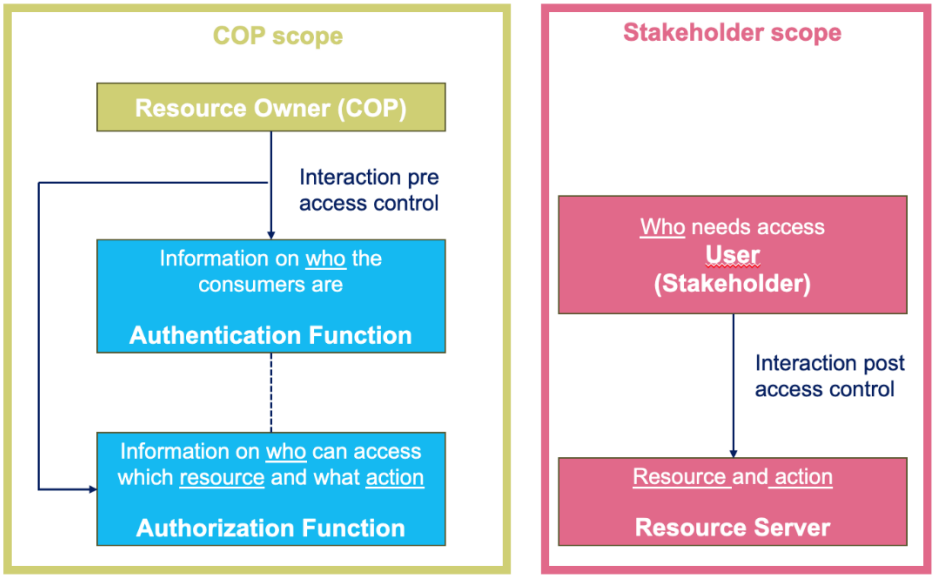

Figure 2-14: Access control governance mechanisms.

2.1.3.2 User-centric service provisioning

The User-centric Service Provisioning system is designed to deliver personalised services tailored to specific and customised SLAs for different stakeholders. This system emphasises the integration of security and privacy considerations, ensuring that user trust and confidence are integral to the service experience.

In the context of 6G networks, the user-centric service provisioning system aims to enhance user experience, optimise network resources, and support the diverse and dynamic requirements of future applications and services. It provides stakeholder subscribers, such as enterprise users and end-users, with an optimal and personalised Quality of Experience (QoE) based on their preferences, SLAs of subscribed services, and the network context. The system can achieve this through several key capabilities.

For this system, customised service policies can be formulated based on the User Equipment Route Selection Policy (URSP) concept defined in the 3GPP standards (3GPP TS 23.503 [23.503] and 3GPP TS 24.526 [24.526]), which are executed by mobile modems in user devices, ensuring service fulfilment. Additionally, the system would guarantee secure access to subscribed services and protect user data by injecting URSP rules into the user devices, which is essential for the services activation. Ensuring compliance with the KPIs defined in the SLAs throughout the service lifecycle, the system would employ closed control loop automation features to correct any violations, targeting the service assurance.

The User-centric Service Provisioning system is closely related to the concepts of intents, SLAs, and closed loops. Intents would be translated into SLAs, breaking down high-level goals into specific, measurable requirements. SLAs would provide benchmarks that CLs would strive to maintain. CLs would continuously monitor performance metrics and make real-time adjustments to ensure that the service meets the SLA criteria. This hierarchical relationship from intents to SLAs to CLs would allow the systematic management of resources and service quality, enabling rapid identification and rectification of any deviations from performance thresholds, thereby maintaining operational stability and performance consistency.

Key components of the system include the intra-stakeholder space and the so-called imported models. The intra-stakeholder space would manage the space provisioned to each registered stakeholder, capturing the URSP rules applicable to individual stakeholder users and injecting these rules into mobile user devices for service fulfilment and activation. Imported models would store models for SLAs and closed control loops, supporting zero-touch solutions for service assurance using CL-based automation.

The internal architecture of this system (Figure 2-15) involves several steps to ensure the seamless functioning of the system. The Digital Service Provider (DSP) [HEX223-D22] [HEX224-D63] sends relevant information from the 3P profiling block[5] to the COP which would allocate it within the trustworthy 3P service provisioning environment. The URSP rules, stored within the intra-stakeholder space, would be interpreted and executed by the devices.

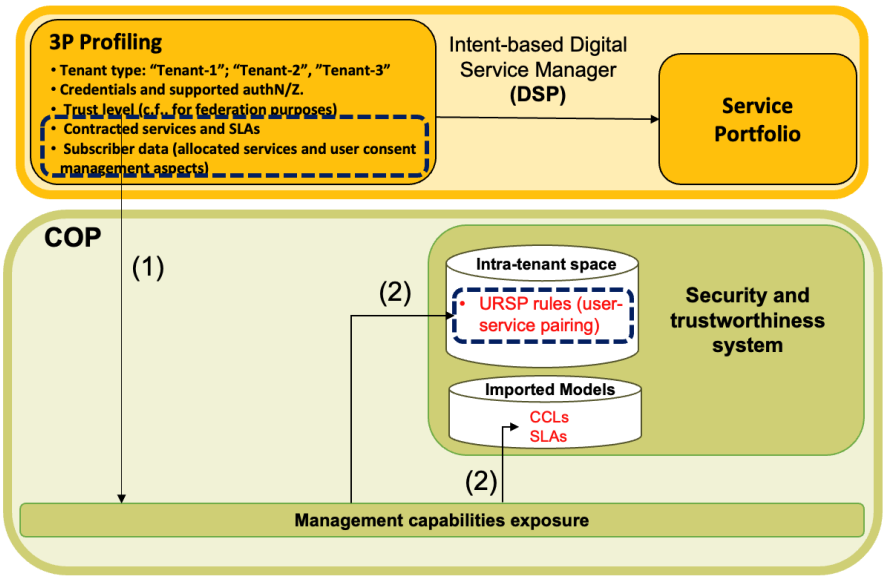

Figure 2-15: User-centric service provisioning – internal architecture.

For service activation, URSP rules would be provisioned to stakeholder user devices. The mobile network would identify which URSP rules to inject into which devices and would forward this information through the subscriber database to the control plane network functions, responsible for injecting applicable URSP rules into individual devices. This process would enable the application client to issue service requests and establish connectivity according to the registered SLAs.

During the service assurance stage, CL instances would be used to supervise the state of the service instance and ensure its conformity regarding the SLA. The system would insert SLA attributes into the CL model, which would become the goal that the CL instance must fulfil and assure. Communication would be performed through the Management Capabilities Exposure functionality, which would request the creation of CL class instances for every contracted service, ensuring continuous compliance with SLA requirements.

2.1.3.3 Network Digital Twins Creation

Introducing AI models into network and service M&O allows to further optimise decisions and provide them in near real-time or even proactively. However, applying these decisions on the actual network infrastructure directly carries the risk of negatively impacting the network and the deployed services if the model is not properly trained on certain scenarios, or if its performance degrades. On the other hand, the Digital Twin concept has gained attention and is being applied to different domains by providing digital replicas of physical entities, which accurately simulate their characteristics and behaviour.

In this context, Network Digital Twins (NDTs) have emerged as a tool to enable safe AI/ML model orchestration and control by generating "what-if" scenarios for training, or for applying orchestration actions and observing their effects before carrying them out on the production environment. In line with this, the aim of this NDTs Creation system is to develop tools and mechanisms to generate an accurate NDT of the network infrastructure which reflects its behaviour and properties. In particular, properly modelling the effect of virtualisation for network and service performance is being investigated.

Figure 2-16 illustrates the NDTs creation system. The application layer provides input to the M&O related functions on a desired state of the network or service in form of an intent (for example). The M&O then performs the decision process to determine the appropriate action to reach the desired outcome. The action is provided to the Network Digital Twin for testing its effect on the network and feedback is provided back to the M&O to determine whether to enforce the actions on the real network, or if different actions are required.

To provide accurate feedback to the M&O, the Network Digital Twin management creates the Network Digital Twin based on data collected from the real network to accurately reflect the state and behaviour of the network.

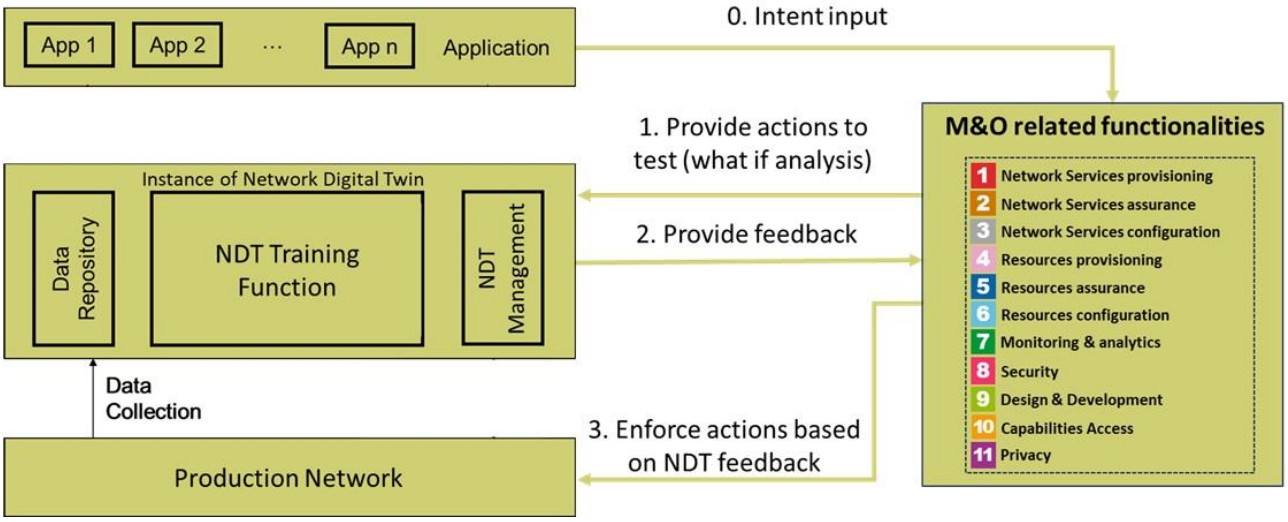

Figure 2-16: Schema of Digital Twin assisted M&O.

The creation of a Network Digital Twin relies on this data to create a model of the infrastructure where the network is deployed. The dynamic nature of virtual networks may require frequent creation and destruction of slices. Neural Network based models have proven to be good in capturing this dynamicity and can represent how an active network state affects QoS behaviour [MAJ+18]. Compared to other architectures (e.g., Deep Neural Networks, Deep Reinforcement learning), Graph Neural Networks (GNNs) seem to outperform the rest in modeling communication networks [SVA+23] mainly because their structure can easily be represented as a graph, whereby the NDTs creation mechanisms proposed here are based on using GNNs. However, many existing mechanisms [FGP+23] to create NDTs do not (i) consider that networks are increasingly virtual and running on a cloud infrastructure and (ii) are susceptible to the problem of long-term dependencies in large-scale and dynamic environments. The method proposed leverages attention mechanisms in graph neural networks and virtualisation-aware modeling that address these gaps.

In order to cater for these aspects of creating NDTs with GNNs, the underlying data collected has to contain information on the impact that Network Functions (NFs) have on each other when they share the same infrastructure resources, including CPUs, Memory, I/O subsystems and Network Interfaces. This will allow any model derived from it to capture the complex non-linear impacts that NFs have on each other. The long-term dependencies, mostly related to the "oversmoothing" effect where node features become indistinguishable [CLL+20] can be solved by using other types of aggregation for neighbour node features rather than the mean. These two features can greatly improve GNNs as a mechanism to create accurate NDTs, which can be used predict performance of large virtual networks deployed on a general cloud infrastructure.

2.1.3.4 Sustainable MLOps

The Sustainable MLOps (S-MLOps) system enables the automation, monitoring, and optimisation of the different workflows involved in the development, deployment, and operation of AI/ML-based network services, considering also the measurement of the energy consumption in the different stages of these workflows across all involved stakeholders in the telco scope. A tangible example of its integration can be found in section 2.2.2.1.

Although MLOps techniques are already being applied in the state-of-the-art, the implementation in the telco-grade scope presents specific challenges, namely:

- Software assets are typically outsourced by the network operators, which towards 6G will require the interaction among different stakeholders to implement the common CI/CD (continuous integration, continuous delivery) workflows in MLOps. This interaction is envisaged essential to maintain seamless model deployment and updates across different domains. Additionally, these communication channels may involve the exchange of sensitive information, including personal user data or confidential business data to train certain AI models. This requires the implementation of robust security measures to protect the integrity and privacy of the data without hindering the training

processes or compromising the performance and efficiency of the workflows. Also, this collaboration among stakeholders often involves the integration of different technologies, tools, and datasets, which requires a highly coordinated effort to ensure interoperability.

- MLOps not only automates the workflows involved in the development and deployment of AI/ML models, but also needs to focus on reducing environmental impact by incorporating sustainability into its practices. Integrating sustainable principles into MLOps processes is considered essential to minimise the carbon footprint. This dual focus on efficiency and sustainability is particularly relevant in the context of 6G smart networks, where the integration of AI/ML capabilities can drive innovation while supporting ecological goals.

The S-MLOps system is an asset envisaged to be used by different parties (e.g., software vendors, mobile network operators…) to create AI/ML-based workflows, giving them also the possibility to monitor and reduce the energy consumption associated to the AI/ML development and deployment processes. This information on energy consumption can be utilised to objectively evaluate the trade-off between system performance and energy usage, facilitating the implementation of energy-saving measures.

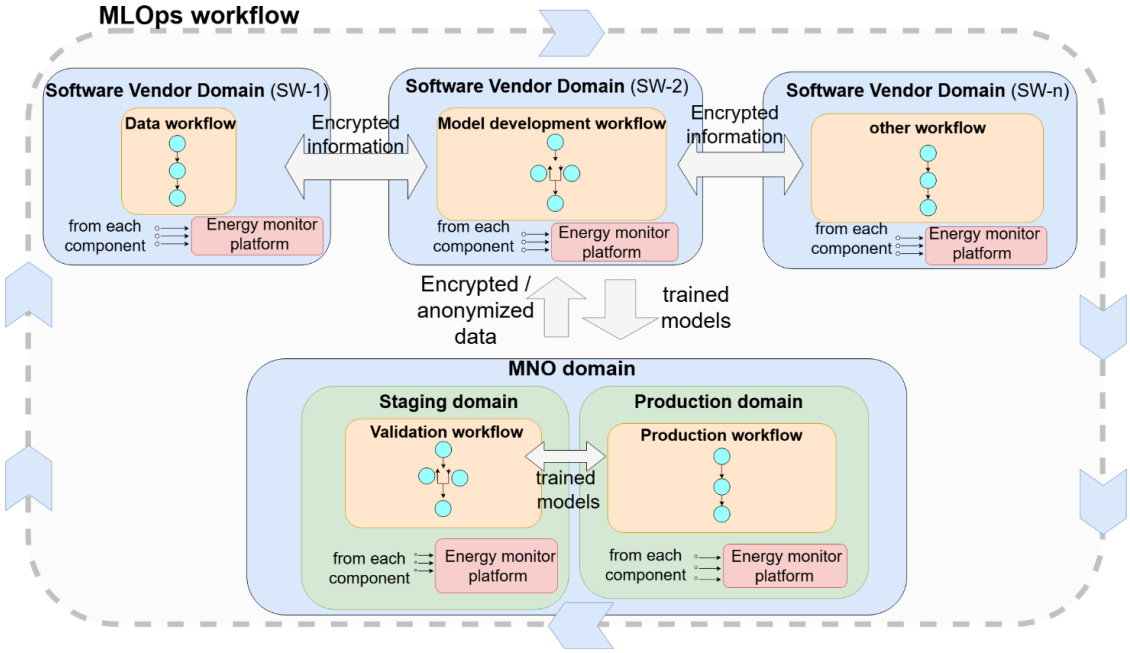

Figure 2-17: Operational scope of sustainable MLOps.

Figure 2-17 provides a general overview of the operational scope of the S-MLOps system. As it can be appreciated, the MLOps workflows can span multiple domains, facilitated by modules that enable the transfer of anonymised and/or encrypted information among them. On one hand, software providers would be responsible for developing AI/ML-based network services, dividing the necessary tasks among themselves. E.g., certain providers may focus on processing data provided by the MNO (data workflow), while others would be tasked with generating AI/ML models (model development workflow). The mobile network operator would receive the trained models for subsequent deployment. This process would typically begin in a staging environment within the MNO scope, where the model would be automatically validated by running predefined validation workflows. If the validation were successful, the model would then be moved to the production environment. The S-MLOps system would enable the creation of workflows for each domain independently, including the deployment of modules for intercommunication and the energy measurement of each stage within the workflows. Based on the energy and performance measurements over different conditions and AI models, and considering the different stages of the workflows, the involved stakeholders can select the most appropriate model prioritizing performance, consumption and sustainability ratios as needed, using information gathered and model sharing APIs as shown in Figure 2-18. Section 2.2.2.2 describes a practical implementation of this S-MLOps system used to deploy an AI/ML component in the context of a proactive service orchestration example.

Figure 2-18: Sustainable measurements and information and model sharing APIs example.

2.1.3.5 Network Programmability

The Network Programmability system can support service providers with a flexible, programmable, and scalable approaches to network control and management. This solution proposed here is based on several key technologies and approaches, including software-defined networking (SDN) [NG13], application programming interfaces (APIs), and cloud-based network management platforms [VMC+21].

The proposed high-level architecture for the Network Programmability system assumes a single administrative domain, and it involves a hierarchical approach with an E2E controller acting as parent controller, and technological domain controllers as child controllers. Specifically, it proposes an E2E SDN orchestrator and technological domain SDN controllers in the IP (Internet Protocol), Optical, and TSN/DetNet and domains [MCT22].

The Network Programmability system, which acts as E2E SDN Orchestrator, is the parent controller responsible for managing and orchestrating the entire transport network infrastructure using Software-Defined Networking. The E2E SDN orchestrator oversees the coordination and control of the network across different technological domains. The E2E SDN orchestrator can rely on existing solutions, such as ETSI TeraFlowSDN (TFS) [TFS24]. TeraFlowSDN and its components has been described at [VMC+21].

As the parent controller, the E2E SDN Orchestrator acts as a unifying entity that harmonises the operations of various network elements, ensuring seamless communication and efficient utilisation of resources. One of the primary objectives of the E2E SDN Orchestrator is to coordinate multi-domain orchestration of network operations. In modern network architectures, different domains, such as data centre networks, wide area networks (WANs), edge networks, and cloud infrastructure, coexist and interact to deliver end-to-end services. The E2E SDN Orchestrator plays a crucial role in integrating and managing network elements from diverse domains. By facilitating the coordination and orchestration of network operations across these domains, the E2E SDN Orchestrator enables consistent policies, optimised resource utilisation, and efficient communication throughout the entire network infrastructure.

Furthermore, the E2E SDN Orchestrator supports service-level orchestration. In addition to managing the network infrastructure, it provides mechanisms for defining and managing services that traverse multiple network domains. By considering the end-to-end service requirements, the E2E SDN Orchestrator ensures that services are provisioned and delivered seamlessly across the network. It enables the dynamic allocation and optimisation of network resources, allowing services to adapt to changing demands and conditions. The connectivity service orchestration capabilities of the E2E SDN Orchestrator enhance the overall efficiency, scalability, and agility of the network, providing a flexible and adaptable infrastructure for delivering a wide range of applications and services, as shown in Figure 2-19.

Technological Domain SDN Controllers are child controllers that operate within specific technological domains of the network. Each technological domain controller would be responsible for managing and controlling the SDN functions and resources within its respective domain. Each specific technological domain controller might be based on several SNS project solutions, for example PREDICT-6G [PRE24] work could be used for DetNet. For IP or Optical SDN controllers, ETSI TeraFlowSDN is also proposed.

Several specific novelties have been introduced in the presented architecture and validation efforts are available in [HEX224-D63]. In Section 2.2.1.6 implementation details regarding this enabler using ETSI TeraFlowSDN [TFS] are provided.

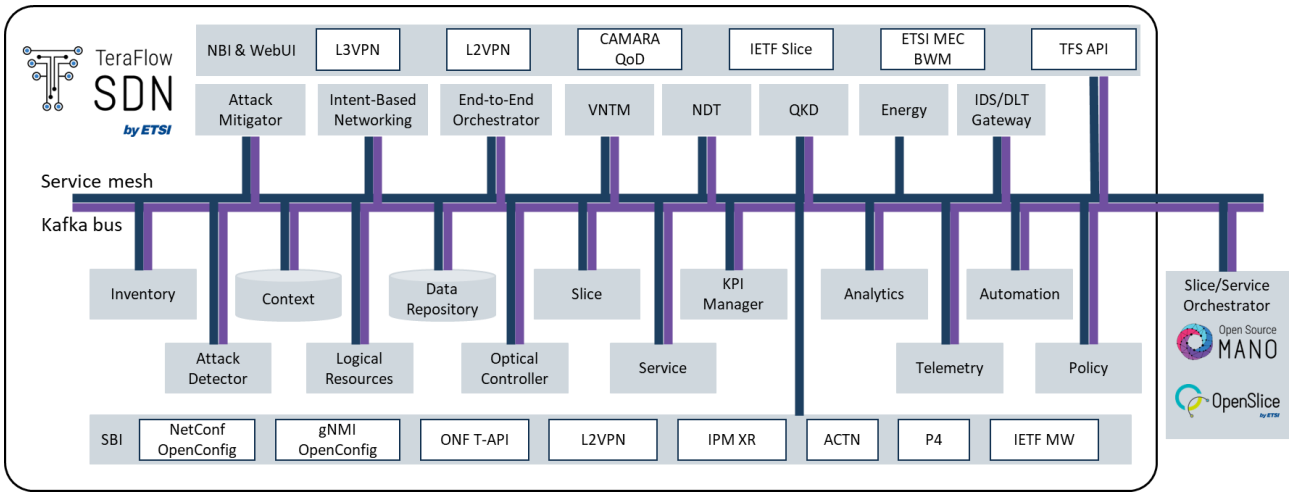

Figure 2-19: Network programmability framework architecture.

2.1.3.6 Privacy Protection for data analytics in M&O

In 5G, the Management Data Analytics Function (MDAF) [28.809] in the M&O layer is responsible to perform data analytics by leveraging current and historical data from the network including data related to RAN, CN (Core Network), TN (Transport Network), as well as data from external entities to provide insights for managing network performance, security, resource allocation, and policy enforcement. Ensuring that this data is handled securely and in compliance with privacy regulations is crucial to uphold trust and ensure compliance. Given the risk of potential attacks, it is essential to protect the 6G analytics functions from processing manipulated or un-sanitised data received from various sources. A protection system that is suitable for various analytics functions and input data is needed to prevent the generated analytics from being affected by the manipulated data. Developing such a system for data analytics in M&O involves implementing various techniques and strategies to ensure that sensitive data is protected while still enabling effective analysis and decision-making.

Since AI/ML is expected to be widely used in 6G, this system is intended to be one of the key enablers used in the future M&O system to orchestrate and optimise network operations. The analytics function (AF) will be one of the functions in M&O that leverage AI/ML to conduct its inference (analytics). Although using AI/ML by this AF enables better optimisation of decisions, it is also crucial to ensure that the data used for training AI/ML models is handled in compliance with privacy regulations (such as GPDR).

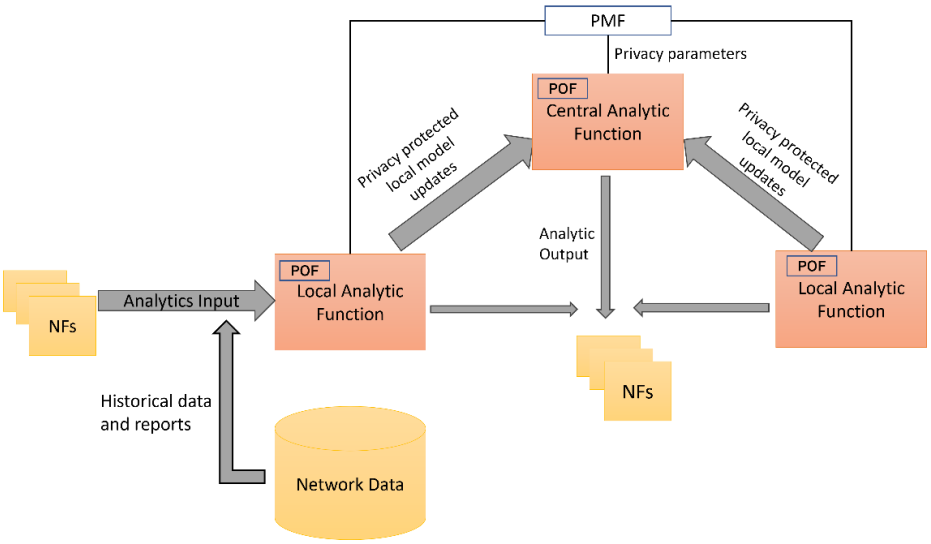

Figure 2-20: Overview of the Privacy Protection Framework for data analytics in M&O.

The analytics outputs that are produced by AF can provide insights to efficiently managing and optimizing network resources, ensuring high-quality service delivery, and enabling intelligent decision. These analytics

outputs would be available to authorised consumers in the form of Network Functions (NFs). Multiple AFs can operate in a network, especially in modern, dynamic network environments. Federated Learning (FL) can be utilised in distributed architectures for both optimisation and privacy. The local model updates in the federated learning from each AF may contain sensitive information that should not be shared with other AFs. However, privacy enhanced federated learning technique could be used to train a global model without disclosing local data of each AF. In this context, adding privacy protection for data analytics in M&O would be beneficial to preserve confidentiality and enhance privacy. For this reason, a Privacy Management Function (PMF) and a Privacy Operation Function (POF) are proposed as illustrated in Figure 2-20, where the POF would be responsible for making the update of the local model privacy-protected, using policies created by the PMF. Beyond that, this PMF would be responsible for selecting the privacy operation (e.g. Homomorphic Encryption and Secure Multi-party Computation), to be used by the POF and the generation and distribution of key pairs for privacy operations, which would take place at local and central AFs in a federated learning setting.

The sequence diagram for the information that are communicated between different components in our proposed framework is illustrated in Figure 2-21. In step 1, the central and local AFs send registration request along with their capability for privacy operation to the PMF. In step 2, according to the received privacy operation capability of the central and local AFs, the PMF will decide on privacy operation, and which AFs to be included in the FL process. In step 3, the PMF registers the central and local AFs and generates a list of clients for FL, along with the required keys and parameters for the selected privacy operation. In step 4, the PMF sends the key and privacy operation to the POF of the central and local AFs which can be incorporated inside or as a separate function outside of central and local AFs. In step 5, the PMF sends FL identifier to central and local AFs. In step 6, local AFs notify the central MDAF that they are ready for the FL operation. The FL process will be started and each local AF performs local training and sends local model update to POF to apply privacy operation, after which the privacy-preserved model updates will be shared with central AF. In this way, it is guaranteed that the central server could not disclose any sensitive data regarding to local data at each AF.

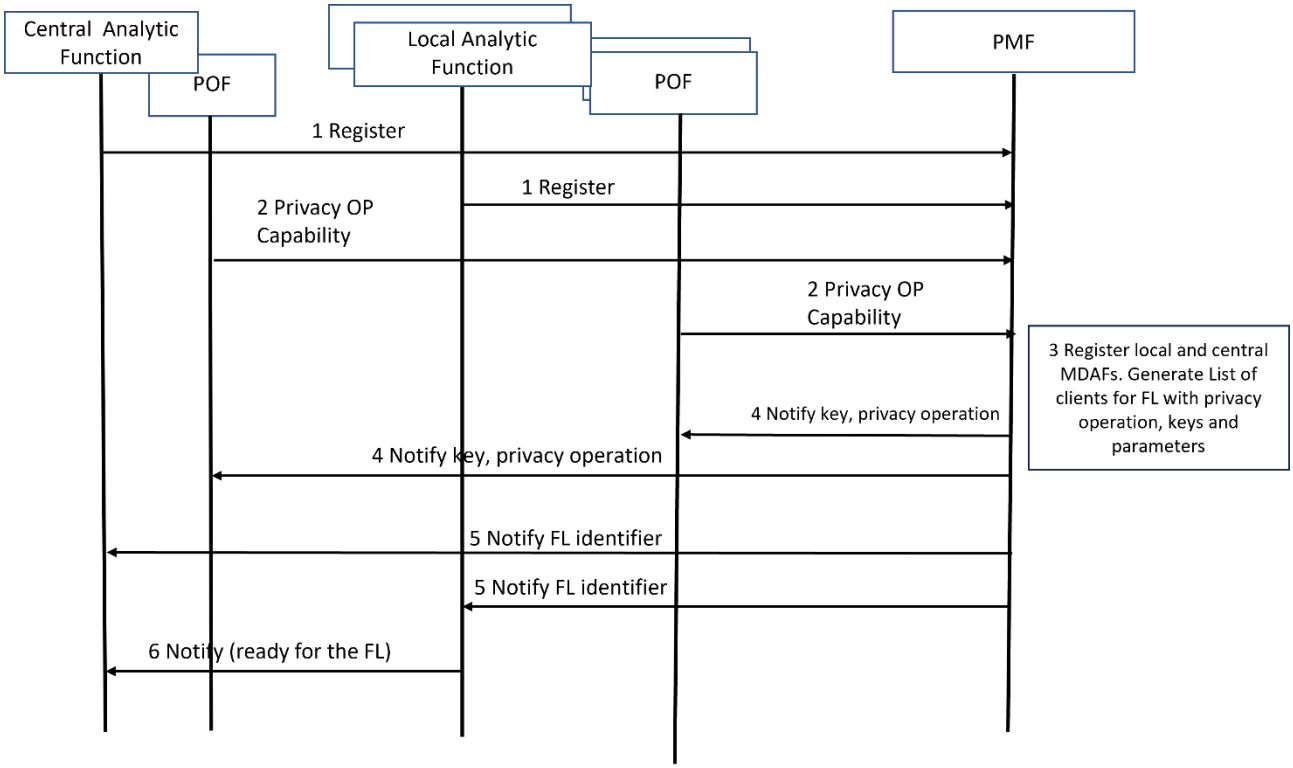

Figure 2-21: Overview of the Privacy Protection Framework for data analytics in M&O.

2.1.3.7 Secure AI/ML-based control for Intent-based Management

The intent-based management (IbM) system being defined in the context of WP2 [HEX224-D24] aims to manage and operate telecommunication systems based on high level and abstract definition of goals and

requirements via intents. By expanding the utilisation of AI/ML techniques, IbMs can orchestrate and optimise network operations autonomously which brings automation and efficiency to the network operations. Although AI/ML models enable the IbM system to dynamically configure network parameters and resources to meet the intents, these models have some intrinsic vulnerabilities against adversarial attacks. Adversarial attacks in intent handling functions may disrupt the network management operations, lead to service interruptions and result in a state where intents are not met. The aim is to identify the potential threats and risks associated with AI/ML-based control in IbM systems, and to provide a secure algorithm to ensure that decisions predicted by AI/ML models are resilient to adversarial attacks and are not tampered with or compromised.

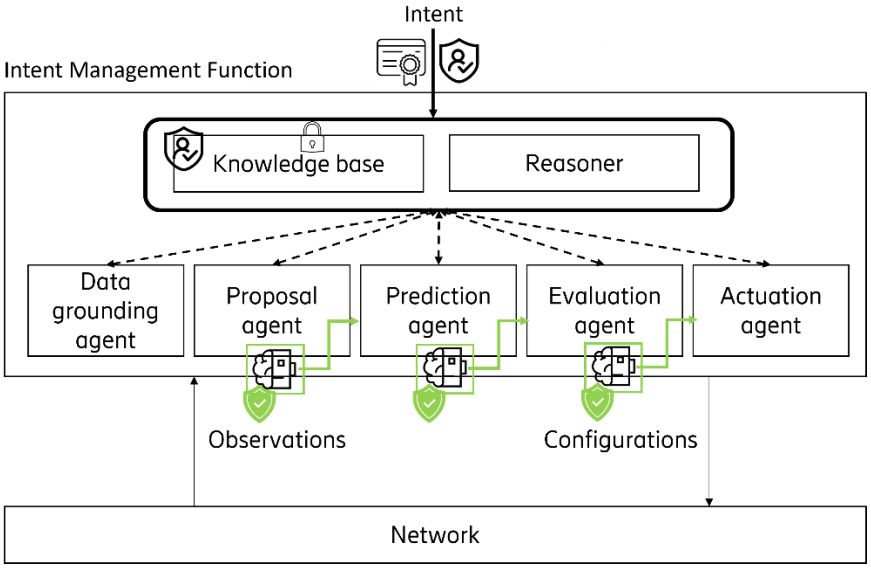

Figure 2-22: Secure AI/ML-based control for the Intent-based Management System.

Figure 2-22 illustrates the secure AI/ML-based control for the Intent-based Management system. As it is shown, there are different agents in the system where each can be deployed with pre-trained ML models. The intent management function follows the decisions which are made by the agents within the closed loop to take an action and reach the desired output. The utilised AI/ML models by the agents need to be robust against adversarial attacks to ensure accuracy and integrity of the actions taken by the system. To ensure robustness of AI/ML models against adversarial attacks, adversarial training techniques are applied by the agents. More detailed information on this topic, can be found in [HEX224-D63] and [KBF+24].

In addition, the knowledge base where all information about network, agents, intent, and where KPIs are stored, need to be secured against unauthorised access. Thus, robust authentication mechanisms (e.g., multi-factor authentication) and fine-grained authorisation policies need to be employed to control who can access and modify the knowledge base. Furthermore, cryptographic techniques might be used to encrypt the sensitive parts of the knowledge in the knowledge base to prevent modification by unauthorised access.

The Reasoner coordinates the closed loop operations to handle the intents and executes the corresponding set of agents according to their roles; so, the integrity of the submitted intents will be important. In this regard, cryptographic techniques such as digital signatures can be used to ensure the integrity of stored and retrieved intents. In addition, unauthorised users might access or modify intents, leading to network misconfigurations or security breaches. Thus, strong authentication mechanisms (e.g., multi-factor authentication) and fine-grained authorisation policies are required to control who can access and modify the intents. Furthermore, encrypting intents at rest using encryption algorithms can be useful to protect them from unauthorised access.

2.1.4 Algorithms

This section outlines the set of AI/ML-based algorithms included within the Hexa-X-II M&O framework, specifically focusing on stakeholder-level scope. Of course, this is not an exhaustive list of the algorithms that could be deployed in the future 6G systems. This block just highlights that the framework must contain a specific Algorithms block targeting certain specific functionalities, although in this case, the focus is just on a

few algorithms that have deemed particularly relevant within the Hexa-X-II project scope, in particular on AI/ML algorithms. They are the following:

- An ML-based configuration recommender algorithm for energy savings, targeting the reduction of the power consumption in the future 6G base stations, intended to contribute to the overall energy efficiency of the network infrastructure.
- A set of algorithms for the efficient allocation of network functions, with a strong emphasis on energy efficiency and targeting also to operate on resource-limited extreme-edge devices and across the entire network continuum, ensuring that the network ecosystem can extend to even the most constrained environments.
- An algorithm for resource assignment in federated learning, targeting to optimally assign compute resources to be used for federated learning with minimal impact on services and minimising energy consumption.
- Multi-Agent Reinforcement Learning (RL) algorithm for adaptive scaling. This is considered a key feature in the framework, offering adaptive scaling of resources. It allows different resource types to collaborate to meet service-level objectives, such as latency and energy consumption, adapting dynamically to changing network conditions.
- Explainability algorithms for RL-based control. The framework also includes an explainability algorithms to provide human-interpretable explanations for decisions made by reinforcement learning-based control algorithms. This enhances transparency and trust in AI-driven decisions, particularly in complex network environments.

The following subsections explain those algorithms and the functionalities they provide with a bit more detail.

2.1.4.1 ML based configuration recommender for energy savings

Despite numerous energy-efficiency solutions already implemented in mobile networks, energy consumption continues to rise due to the rapid growth of network traffic and data volumes [ERE22]. Additional energy savings can be achieved by utilizing ML techniques that enable higher levels of automation. This algorithm considers that ML-based techniques, that can recommend optimised configuration settings for base stations and other equipment, can reduce energy consumption in network elements without affecting the QoE.

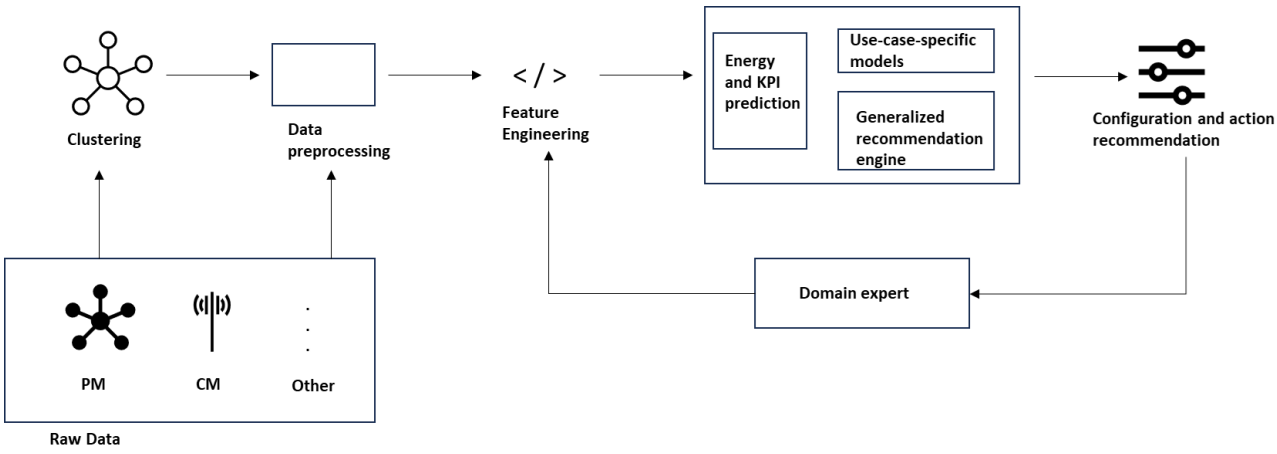

Figure 2-23: End-to-end energy optimisation from power system to node to network.

The efforts in this work to improve energy efficiency focus on access nodes within networks. An access node, often simply called a node, refers to the connection between end-user devices and the network elements they connect to. The configuration settings of these nodes significantly influence their energy consumption and can impact many observable QoS metrics. Since configuration settings at a particular node rarely change, regulating node energy consumption requires a mechanism to generate and evaluate new configuration settings to assess their impact. To avoid configurations that might negatively affect existing KPI levels, new settings need to be constrained by different KPIs that guide the search. Given that some configuration changes may take longer to implement than others, accurate predictive models are needed to anticipate when such changes can be applied in advance. This approach minimises potential disruptions to network operations. An ideal solution would identify as many potential methods as possible to reduce energy consumption without

compromising the current functionality of network elements. Figure 2-23 illustrates the concept, highlighting the energy recommendation engine at its core.

The energy recommendation engine algorithm can generate a set of configuration attribute changes for each corresponding node. Instead of focusing on isolated fine-tuning at the individual node level—which could inadvertently affect other nodes—the output considers the interactions between different nodes and their configurations [VHT+24], [SBN+23]. As a whole, the algorithm proceeds as follows: as a first step, the necessary data that is related to performance KPIs, and configuration management (CM) is collected and then the data is pre-processed and cleaned. Next, the required feature engineering is performed. Later, after determining the most important feature for the objective, a Generative Artificial Intelligence (GenAI)-based solution that uses conditional variational autoencoders is trained, and the solution recommends a new network configuration (i.e., new transmit power level) by considering the current situation of the network and the predicted energy level.

2.1.4.2 Efficient network and service function allocation

Efficiently managing services and workloads in future networks is considered essential for reducing energy consumption. This is particularly important due to the growing number of wireless devices with diverse requirements, and the need to support a range of new, challenging services. The functionality allocation component described in this subsection optimises the placement of computational workloads/tasks to the available compute nodes (e.g., edge/cloud servers, robotic units) with maximum energy efficiency and trustworthiness. Energy efficiency is achieved by minimising computing and transmission energy consumption, and accounting for energy generated from RF harvesters if available.

The functionality allocation optimisation algorithm takes as input the computational workloads/tasks, each with specific requirements and data size and location of data generation, and the capabilities of compute nodes, including their power consumption, battery level, and trust indexes (which are coming from the functionality described in Section 2.1.2.5 and are a function of availability, reliability, security, multi-connectivity capability and battery level if applicable). It also considers the network topology graph with the communication channels. The goal is the minimisation of the multi-objective weighted (w_1, w_2, w_3) objective function $OF = w_1 E + w_2 L - w_3 T$ consisting of the energy consumption E, the E2E latency L and the trustworthiness T term (sum of the trust indexes of the utilised compute nodes for the workload placement). Constraints include each workload being allocated to only one node, respecting node available resources, and fulfilling task requirements. The output is the optimal placement of workloads to the available nodes.

The functionality allocation mechanism can perform resource allocation on compute resources across the continuum, as well as device-specific functionality allocation optimisation. This can be approached by considering the physical tasks/roles with their requirements, such as robotic-specific features, cameras, wheels, propellers, the physical distance for task handling and the various physical nodes, with their capabilities (e.g., available CPU, memory), power consumption, battery level, robotic-specific features, and physical location. The goal is the minimisation of the total energy consumption of the physical nodes, the total travel distance, and the longest tour among robots to ensure equal workload distribution.

In particular, a set of n physical tasks (e.g., robotic tasks/roles) is assumed for approaching the physical task planning problem, with specific functional requirements and the physical distance matrix $D = (d_{i,i'})$ consisting of the distances between the physical tasks. Also, a set of m physical nodes e.g., robotic units, is assumed, with their capabilities. These are the battery level, the physical location, the status, and the availability of wheels, propellers, camera and so on.

The aim is to plan the physical tasks to the available physical nodes (propose a sequence of tasks per physical node) and ensure efficiency of the system with low energy consumption and latency. The multi-objective function which is minimised is:

$$\min_{x,y,maxD} \left(a_1 \sum_{j=1}^{m} y_j \cdot p_j + a_2 \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{i'=1}^{n} x_{i,i',j} \left(\frac{tw_j \cdot d_{i,i'}}{u_j} + g_{i'} \cdot dw_j \right) + a_3 \cdot \text{maxD} \right)$$

The first term denotes the sum of the costs p_j associated with utilising each physical node in relation to a

potential minimum resources' intent. y_j is a binary decision variable that indicates if physical node j is utilised or not. The second term denotes the total energy consumption where $x_{i,i',j}$ is a binary decision variable indicating if physical node j is aimed to execute task i and then task i', tw_j and dw_j are the power consumption of robot j when traveling and when doing a task, respectively, u_j is the mean velocity of robot j and $g_{i'}$ is the duration time of completing task i'. The third term is related to the maximum time duration a utilised robot has, where maxD is the maximum time of a subtour decision variable. The terms of the OF are normalised and weighted (a_1, a_2, a_3) depending on the use case.

The constraints of the problem are:

- $\sum_{j=1}^{m} z_{i,j} = 1$, $\forall\, i = \{1, \dots, n\}$ all physical tasks should be handled by only one physical node ($z_{i,j}$ is a binary decision variable indicating if physical node j handles task i).
- $\sum_{i'=1,i'\neq i}^{n} w_{i,i'} \leq 1$, $\forall\, i = \{1, \dots, n\}$, each task should be ended only once or be the last to be ended (when task is location, leave each location only once or stay there as final destination-no need to return to the initial position). $w_{i,i'}$ is a binary decision variable indicating if task i' is handled after tasks i.
- $\sum_{i'=1,i'\neq i}^{n} w_{i',i} = 1$, $\forall\, i = \{1, \dots, n\}$, each task should be executed only once
- $\sum_{j=1}^{m} x_{i,i',j} \leq 1$, $\forall\, i, i' = \{1, \dots, n\}, i \neq i'$, the execution of task i'after the task i should be made by only one physical node.
- $\sum_{i=1}^{n} z_{i,j} = \sum_{i=1}^{n} \sum_{i'=1}^{n} x_{i,i',j} + 1$, $\forall\, j = \{1, \dots, m\}$, the number of tasks handled by a physical node equals the number of edges a physical node passes plus one.
- $z_{i,j} \leq c_{i,j}$, $\forall\, i = \{1, \dots, n\}\ and\ j = \{1, \dots, m\}$, the feasibility of assigning a physical node to a task in terms of functionality types that are supported (e.g., having propellers or camera), is respected. $c_{i,j}$ is 1 if physical node j has the necessary functionality to handle task i.
- $\sum_{i=1}^{n} z_{i,j} \leq b_j$, $\forall\, j = \{1, \dots, m\}$, each physical node can handle up to b_j tasks based on the battery level.
- $x_{i,i',j} \leq w_{i,i'},\ x_{i,i',j} \leq z_{i,j},\ x_{i,i',j} \leq z_{i',j},\ x_{i,i',j} \geq w_{i,i'} + z_{i,j} + z_{i',j} - 2\ \ \forall\, i, i' = \{1, \dots, n\},\ j = \{1, \dots, m\}$ constraint for linearising the variable $x_{i,i',j} = w_{i,i'} \cdot z_{i,j} \cdot z_{i',j}$
- $k_{i,j} - (n+1) \cdot x_{i,i',j} \geq k_{i',j} - n,$, $\forall\, i, i' = \{1, \dots, n\},\ i \neq i'\ and\ j = \{1, \dots, m\}$, modification of the subtour elimination constraint (Miller-Tucker-Zemlin) suitable for this case. $k_{i,j}$ is a continuous variable for having different sequential id in the planned route.
- $\sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{i'=1}^{n} x_{i,i',j}\, d_{i,i'}/u_j \leq \text{maxD}$, $\forall\, j = \{1, \dots, m\}$, the total time duration of the subtour of each physical node should be less or equal to the maximum time of a subtour decision variable (maxD).

Physical task planning is a highly complex problem and using traditional optimisation algorithms to solve it may become time-consuming in large scales. Hence, this problem was solved with the development of a metaheuristic algorithm based on the ant colony optimisation algorithm (ACO) [DSS04]. In the proposed metaheuristic algorithm, each ant of each colony represents a global solution of the physical task planning problem. During the plan (route) construction the different colonies (which consists of a set of global solutions-ants) share experience through pheromone levels. The pheromone levels are stored in a matrix (initially set to uniform) same size as the physical distance matrix and are updated by the end of each colony based on the best solution/paths found by the ants. A set of global solutions is reached for all colonies, but the best solution is selected after the last iteration which is defined by a dynamic stopping criterion. Each physical node (robot) move/picks next task considering the pheromone levels and the restrictions based on robot type (AGV, UAV) and capabilities. This implementation and results are presented in Section 2.2.2.3.

In a different solution, resource efficient network function deployment for Edge scenarios is considered, where Edge Computing nodes are used to deploy services in a de-centralised manner, closer to the end users. This allows to reduce the communication latency, reduces load from the network core, and enables a more efficient resource management. However, Edge Computing resources are limited, which means that an efficient resource allocation mechanism is needed to distribute the network and service functions over the resource-limited Edge servers, while satisfying the SLA requirements. In this case, energy and resource efficiency can

be achieved through consolidation by deploying virtual functions on a limited number of Edge servers to reduce energy usage.

To this end, an intelligent resource orchestration algorithm is proposed to perform the placement of network and service functions and user demand distribution over the functions where the AI/ML model uses monitoring data from the network to provide actions to be carried out by the M&O.

To make decisions, the developed orchestration algorithm employs a Deep Reinforcement Learning (DRL) model which has been trained using the Deep Deterministic Policy Gradient (DDPG) algorithm [LHP+15], which is an actor-critic algorithm that comprises an actor for providing actions, and a critic for evaluating the decisions provided by the actor. The goal is to perform placement and user demand distribution in a way that satisfies SLA requirements in terms of latency and packet loss, while minimizing resource usage by consolidating functions on the same servers whenever possible.

2.1.4.3 Multi-domain federated learning

The compute infrastructure of 6G networks is envisioned to be even more disaggregated, as compared to current deployments, creating a continuum from the edge to the cloud. Given the considerable CAPEX in deploying compute infrastructure at the network edge, close to the base stations, telco operators will need to leverage these resources to provide advanced services. A key example of this is Compute-as-a-Service (CaaS) enabling the training of complex AI models which is currently only feasible with large cloud providers. Unlike the latter, telco operators have access to rich data sources stemming from the plethora of smart devices that rely on them for connectivity. Given the privacy implications these data must be exploited at the nearest compute resources at the network edge to avoid leaks and security breaches. The foregoing naturally implies that distributed learning must be implemented to deliver this AI training service.

Given the constraints imposed by the QoS of legacy network services, the choice of which data sources (and consequently which compute resources) should be used to train an AI model is non-trivial. Therefore, the M&O system needs to be enhanced with algorithms to optimally assign compute resources to be used for the decentralised learning with minimal impact on other services and with the least energy consumption footprint.

Decentralised Learning (DL) trains models across multiple entities, collecting data locally, and aggregating models from each entity. However, DL faces the challenge of accurately identifying and integrating relevant models, as errors can lead to inaccurate aggregated models. Recently, an E-TREE DL architecture has been proposed [YLC+21] that organises nodes into clusters, enabling hierarchical aggregation of model weights.

This optimised decentralised learning technique enhances ML training performance within the MLOps management function. An administrative domain, consisting of nodes with distinct data sources and compute resources, can be managed individually, but interacts to share model weights and produce a local model.

In the proposed system, multiple administrative domains collaborate within a Federated Domain Set (FDS) to optimise shared resource utilisation. This interconnection is managed by an End-to-End Federation Layer, overseen by a Federating Orchestrator (FO) in each domain. Each domain maintains training and testing datasets, with the testing dataset dynamically updated to reflect current data distribution and accommodate potential data drift.

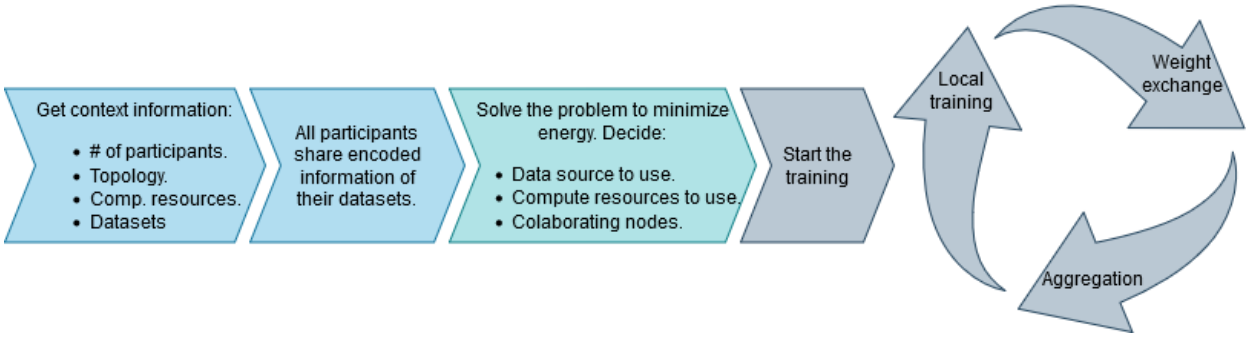

Figure 2-24: Resource assignment algorithm for decentralised edge learning.

The steps involved in selecting which resources to use in carrying out decentralised learning across multiple domains is shown in Figure 2-24. The algorithm takes as input the network topology with all its compute,

storage, as well as their corresponding connections. The relative importance of each source is calculated and the most distinct ones are chosen based on how close and well connected they are to the compute resources. The algorithm also takes into account the connections that exist among the compute resources to optimise the formation of clusters to facilitate the sharing of weights and model aggregation.

2.1.4.4. Multi-agent Reinforcement Learning for adaptive scaling

The increased complexity of network services and application graphs that could be deployed across the network continuum does not fit well with centralised approaches such as optimisation techniques or single-agent-based AI algorithms, whose decision space grows exponentially with the addition of services or links. Handling resources or tasks in a decentralised manner becomes more and more crucial for the satisfaction of the service level objectives (SLOs) defined in a set of available deployment options. The aim of this algorithmic approach is to provide a framework of responsibility distribution across the service or infrastructure domain, defining the agents and the corresponding items they manage towards performing adaptive scaling to optimise efficiency. This includes homogeneous agent groups such as agents for each service, computing or network node, as well as heterogeneous groups that allow different types of resources to efficiently work together to satisfy heterogenous SLOs (e.g. service latency and CPU consumption).

For realizing such a scenario, a Directed Acyclic Graph (DAG) -formed application graph with individual services communicating with each other- is deployed in a multi-provider setup. The traffic created by user requests is distributed across the graph according to the application's structure and so, each service needs to handle different loads at each time point. The developed autoscaling mechanisms make the application adaptive to the variable incoming workload by spawning the necessary number of replicas for each service to handle the corresponding load, while considering the defined optimisation objectives, i.e. latency and energy requirements, and the respective resource constraints.

For distributing decision-making across different entities, a multi-agent approach is considered, as shown in an example in Figure 2-25, where different services are handled by their individual agents. In this instance of the multi-agent mechanism, the agents share a common reward function with their main objective being the common end-to-end latency of the depicted service chain (i.e., service 1 → service 2 → service 3).

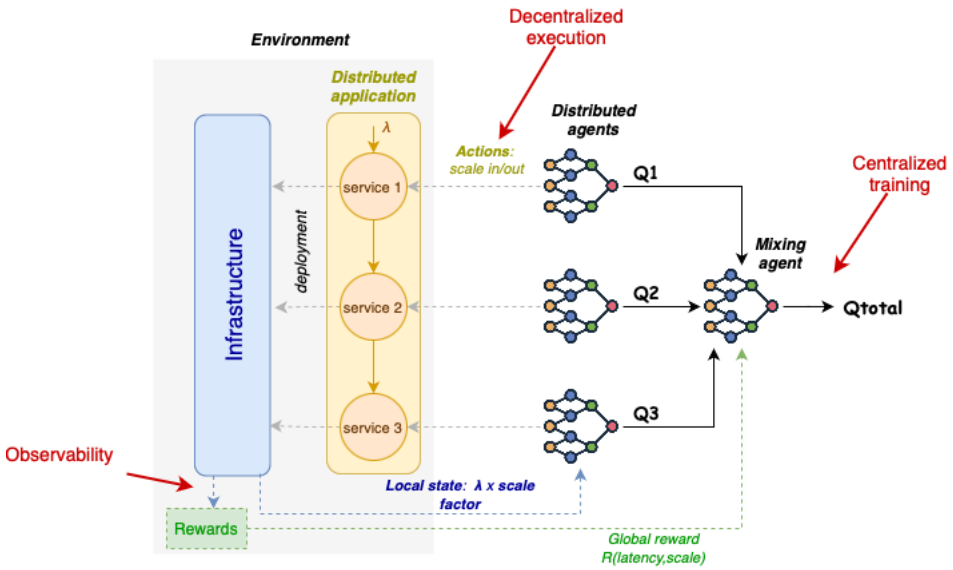

Figure 2-25: Multi-agent setting for autoscaling.

In this instance, the algorithm is formulated to optimise towards the total latency of the requests and the overall resource consumption, while each agent takes responsibility for the monitoring of each service's metrics and its deployment's configuration for satisfying the given constraints. As illustrated in the figure, each agent monitors the metrics for its own service and takes its own actions based on its model, but due to the nature of the specific agent distribution, a collaborative approach is selected for the training process aiming to optimise towards a common reward. For this purpose, the QMIX [RSS+18] variation is selected for enforcing policies that enable global latency minimisation instead of local optimisation, which takes place in cases where

collaboration strategies are not applicable (e.g., independent RL for managing agents for non-communicating services). In cases where there is not a global reward, but collaboration can still improve local rewards (e.g. for disaggregated services that use a common resource pool), decentralised/message-passing approaches [CCK20] can be considered for enabling the understanding of the agents' working environment.

2.1.4.5 Explainability for RL-based Control

Network complexity will increase dramatically in the envisioned 6G system, requiring advanced automation to manage and optimise operations. Reinforcement Learning (RL), a form of AI/ML capable of closed-loop control, has emerged as a promising solution for network optimisation, outperforming traditional rule-based methods that cannot adapt to real-time changes. In 6G network optimisation, multiple agents may control different components of the system. For example, each antenna in a RAN may be controlled by a group of RL agents, each controlling a parameter such as electrical tilt angle and transmit power. Therefore, the actions of such RL agents correspond to adjustment proposals for network re-configurations. However, state-of-the-art RL models operate as black boxes, where the internal decision-making process remains opaque. This lack of transparency makes it difficult for engineers and network operators to understand the rationale behind the actions taken by RL agents. Without clear explanations for why certain decisions are made, the trustworthiness of RL-based automation is limited, especially concerning reliability and accountability.

To address these issues, eXplainable Reinforcement Learning (XRL) techniques can be employed to enhance the interpretability of RL-based decision-making. The proposal here relies on using *post-hoc* explainability [PV20] that, as shown in Figure 2-26, introduces additional components, the *explainers*, without impacting the original RL agents. Thus, in contrast with intrinsic explainability, which modifies the RL agents, this approach enables explainability while using the latest state-of-the-art RL algorithms. The explainers collect and analyse data from RL agents to produce explanations. These explanations can be consumed either by automated validation procedures or by network operators and AI engineers to interpret the reasoning behind RL-driven decisions.

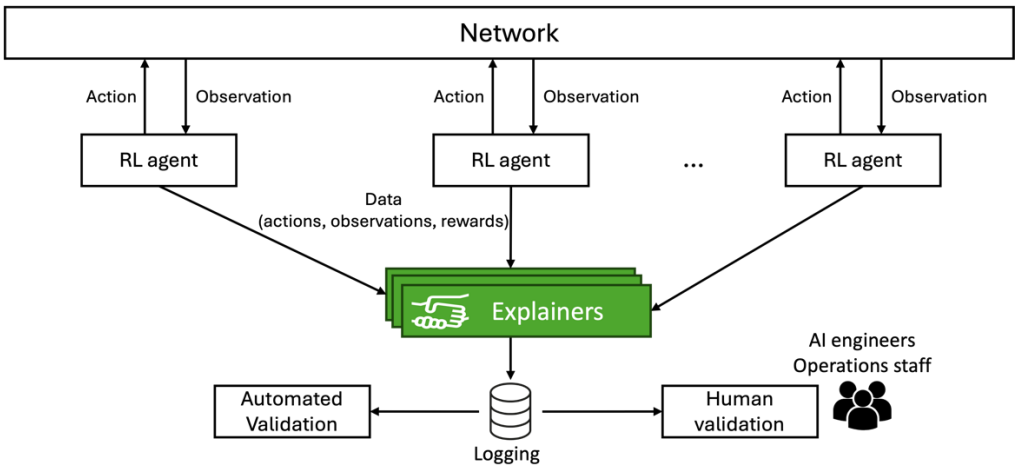

Figure 2-26: Explainable Reinforcement Learning for Network Optimisation.

The proposal is employing a combination of XRL methods to generate explanations at different levels of abstraction, granularity, and perspectives. Focusing on individual agents separately, a set of explainers provide agent-level explanations answering questions such as "Why did you take this action?". This per-agent explainability can be achieved, (i) by using feature importance methods that, for an action proposal in a certain network state, clarify the contribution of each feature in the network state [TIB+20, TIF22], (ii) by learning auxiliary value functions that provide insights into the intended future outcomes that an agent tries to pursue with an action [YRH20,WDB+18], and (iii) by summarizing the agent's policy in natural language [HS17]. Focusing on groups of agents, instead, another set of explainers can generate explanations at a higher level to identify the role and contribution of each agent in the system. The goal, in this case, is to monitor the system and possibly identify suboptimal parts that require deeper investigation. This type of system-level explainability can be achieved through Shapley values for multi-agent RL, where each agent represents a

player in a coalition [RET+24]. This comprehensible approach aims to provide a holistic understanding of the RL-based system, enhancing its interpretability and facilitating trust in its decision-making.

2.2 Implementations

As explained at the beginning of this Section 2, the objective of the WP6 Smart Management Framework is to serve as reference for the design of the E2E architecture that is being addressed in WP2, and also, to serve as benchmark for the design of the future true 6G systems in what regards their M&O capabilities. To fulfil this, as it can be seen in the previous sections, the framework is made up of a wide diversity of components and systems, targeting different functionalities in different network scopes. However, this does not mean that each and every component of the framework must be implemented. Instead, what the framework offers is a variety of features intended to provide 6G system designers and developers with the flexibility and freedom to create customised implementations to meet their specific requirements.

Based on this, it is expected that designers and developers make a needs and requirements assessment, considering what problems need to be solved or what specific functionalities need to be in place. From that, a selection of the specific enablers in the framework would be made, combining those functionalities that were considered the most appropriate. Besides, this flexible approach makes it possible to iterate and adapt the design, adding or changing enablers if new needs or issues should be addressed.

Following this approach, different implementation examples are provided. Some of them target the implementation of certain framework enablers themselves (in Section 2.2.1), while others target how the framework could be used as a whole to implement certain specific functionalities (Section 2.2.2).

Figure 2-27 provides the list of considered KPIs for the performance evaluation of the components of the Smart Network Management Framework, along with the specification of the targeted KPIs per component. Per KPI, a set of metrics are defined for its assessment, where the metrics are quantifiable and can be associated with target values. The full list of metrics considered for the assessment of each KPI per component is detailed in Annex A.4.

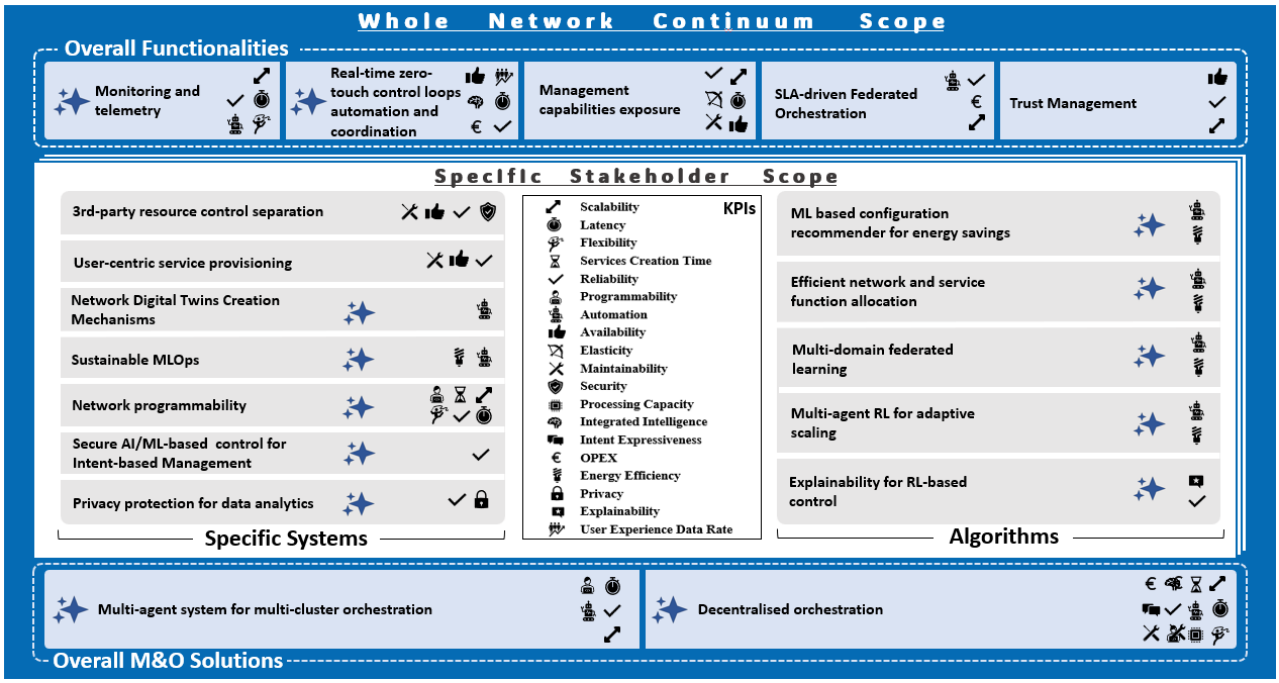

Figure 2-27: List of considered main impacted KPIs per component in the framework.

2.2.1 Framework components implementation

The previous Deliverable D6.3 [HEX224-D63] already provided details and suggestions about how to address the implementation of certain enablers that have been now included in the management framework presented in this D6.5. Specifically, implementations for the following components were already provided:

- The Network Programmability specific system, which implementation (based on TeraFlowSDN) was described in Section 3.1.2 in D6.3.
- The Monitoring and Telemetry functionality, for which different implementations were suggested in Section 3.2.2 in D6.3.
- The Management Capabilities Exposure overall functionality, for which a specific implementation was introduced in Section 3.3.2 in D6.3.
- The Trust Management functionality, with a preliminary implementation and early validation results described presented in Section 3.4.3.2 in D6.3.
- The Multi-agent System for Multi-cluster and the Decentralised overall M&O solution, with different related implementations described in Sections 3.5.1.2 and 3.5.2.2 in D6.3.
- Early implementations of the following algorithms were also provided under Sections 3.6.1.2 and 3.6.2.2 in D6.3:
    - The ML based configuration recommender for energy savings algorithm.
    - Certain efficient network and service function allocation algorithms.
    - Multi-domain federated learning algorithms.
    - Multi-agent RL for adaptive scaling algorithms.
    - Secure AI/ML-based control for Intent-based Management algorithm.
- The Network Digital Twins Creation Mechanisms system, with related implementations described in Section 3.7.2 in D6.3.
- The overall Real-time Zero-touch Control Loops Automation and Coordination functionality, reported in Section 3.8.2 in D6.3, and integrated now as an overall functionality in the management framework.

Beyond this initial work reported in D6.3, new implementations for the framework components have been also performed and tested since that previous deliverable was released. They are described in the following subsections.

2.2.1.1 AI-enabled Real-Time zero-touch control loop analysis function

The implementation of an AI-enabled Real-Time (RT) zero-touch control loop analysis function targets the RT zero-touch control loops automation in the Smart Network Management Framework, to aide in the autonomous control and orchestration in complex networks. The cloud-native, AI-enabled analysis closed-loop (CL) function is one of the four stages of a generalised CL as described in Section 2.1.2.2 with the aim to deliver intelligent insight to the knowledge base of a CL by making inference-based service migration scores from metrics taken by the monitoring stage of the CL of the managed entities. The function has been developed using a containerised micro-service approach, currently using an MQTT [MQT] communication fabric to interface with the other CL stages, with the ability to be adapted to the MCE functionality in the management framework. The function supports integration with a CL Governance service for configuration and provisioning at time-of-service instantiation.

There are several advantages of this analysis function implementation. The containerised application of the function allows for deployment in an environment-agnostic manner, and the microservice approach allows incremental updates, both to the function as well as to the model with minimal service interruptions. As an example, a running analysis CL function instance can be reconfigured with a new model version that has been trained, tested, and deployed without interrupting the closed-loop instance service. The flexible communication interface allows for the addition of model performance monitoring to track model drift, and the use of an Adaptive Neuro Fuzzy Inference System (ANFIS) ML model [JAN93] for inference of the managed entity broadens the capabilities of the closed-loop instance for complex, multi-source data inputs, where analytical representations of the system are not possible or cost-effective.

Two relevant workflows associated with the incorporation of the AI-enabled analysis function into a CL instance are described in Section 2.1.2.2 and include the provisioning of the analysis function as a stage in a CL instance, and the runtime operation inside a closed-loop instance during runtime. Figure 2-28 shows a detailed view of the operation of the analysis function during runtime.

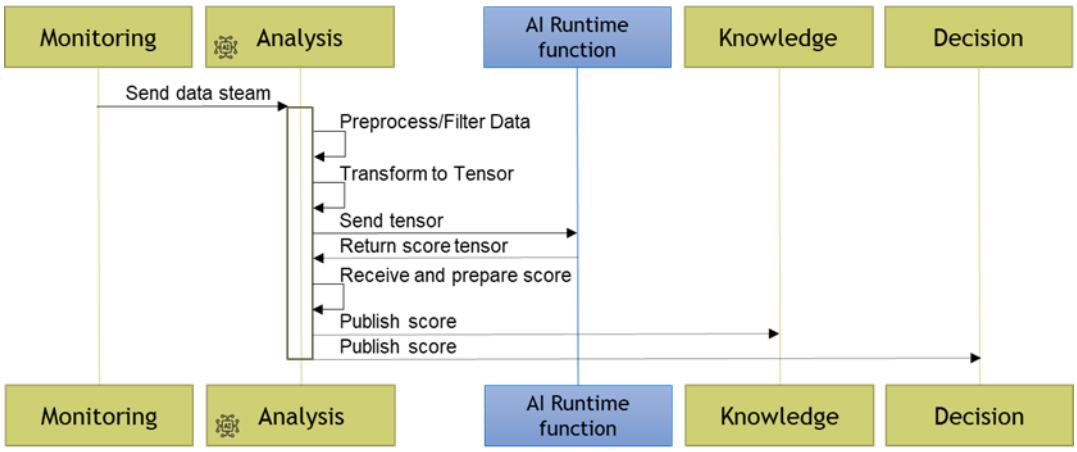

Figure 2-28: Workflow of the AI-enabled analysis CL function at runtime.

At runtime, a Monitoring CL function collects and sends real-time data which is the raw input for the analysis process. The raw data undergoes preprocessing including filtering irrelevant information (e.g., NaN values) before it is transformed into a structured tensor format which is compatible with the model signature of the used machine learning model. The created tensor is then sent to an AI Runtime function which is responsible for generating a score tensor which is returned to the Analysis CL function. The score is then processed and sent to the Knowledge entity of the CL instance which updates its knowledge base, and a Decision CL function which uses it to make decisions or trigger specific actions in the system.

The ML model used for validation in this implementation has been developed following the ANFIS [JAN93] environment shown in Figure 2-29. The ANFIS environment was chosen based on its ability to act as a universal estimator applicable for telecommunications applications [TKC23][OGB18][SSB18], to give insight score (γ_{output}) on the managed entity based on input monitoring data (X_1, X_2, ... X_n). ANFIS integrates both neural networks and fuzzy logic principles to produce a robust yet flexible inference system. Each layer of the environment's neural network corresponds to a step in the ANFIS. The first layer, L1, contains nodes that represent the membership scores (P_{mn}) generated based on the values of the fuzzy input variables. In Fuzzy Logic, memberships represent the discrete classes which an input can belong (e.g, low, medium, high). Although the number of memberships is discrete, the degree of membership for an input is continuous and not restricted to a single membership. The continuous and piecewise differentiable functions implemented as options for this layer include Gaussian, bell-shaped, sigmoid, and triangular. In the second layer, L2, each node represents the accumulated firing strength of rule antecedents, that is, the condition that needs to be satisfied for a rule to be triggered. Rules in this context are analogous to if/then statements, however the consequents are not crisp (e.g. true/false), but rather can be true to a degree, instead of entirely true or entirely false. In the third layer, L3, each node outputs a normalised firing strength, and in the fourth layer, L4, each rule consequent is calculated with associated parameters. Finally, the single node in the fifth layer, L5, can be expressed as the sum of the linear combinations of consequent outputs.

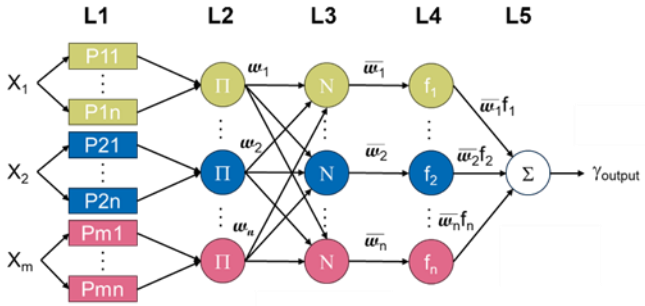

Figure 2-29: Adaptive Neuro Fuzzy Inference System (ANFIS) environment.

An initial validation of this implementation was done as part of a PoC#B.1 with the goal to dynamically and optimally allocate virtual edge compute resources, migrating a deployed service from one managed entity to another based on analysis of ever-changing states of the resources to reduce downtime for a specific service.

The trigger for this is based on forecasted states of the User Equipment (UE) cobots and the requirements of the services.

The overall workflow of the closed-loop at runtime for the PoC with video surveillance cobots is as follows:

1. A monitoring CL function provides streaming metric collection from the cobots, including position and battery level;
2. The position and battery level are used as input to the ANFIS model which provides a Migration Trigger Score;
3. This score along with historical scores is evaluated by a Decision CL function to decide when to migrate the service to reduce service downtime caused by the mobile nature of the cobots;
4. The Execution CL function triggers the migration through an action request at the level of the resource orchestrator, based on the decision made in the system.

The analysis function of the ANFIS model is designed to act as a universal estimator, applicable to a wide variety of optimisation scenarios. For evaluation, the model was trained using labelled historical data from the PoC#B.1 cobot testbed of the continual stream of the cobots' positions and battery levels as they perform mobile video surveillance monitoring around the testbed area. For this scenario the goal of the ANFIS was to learn to determine the scale from -1 to 1 of the estimated efficiency gain from service migration based on cobot battery level and relative position. The data used for training and evaluation was taken from the PoC#B.1 cobots in a period spanning 6.5 hours, roughly equalling 342 complete circuits around the testbed area. Figure 2-30 shows the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 obtained for training and validation datasets for the 4 different membership function types (Gaussian, bell-shaped, sigmoidal, and triangular). For each membership type, the number of membership functions, n, per metric was evaluated for n = 2, 3 and 4. For each evaluation, 50 epochs were used with a batch size of 64. The results of the evaluation show that for this case, the choice of a triangular membership type slightly outperforms when the number of memberships is greater than 2. However, all membership configurations were able to achieve adequate results for this dataset.

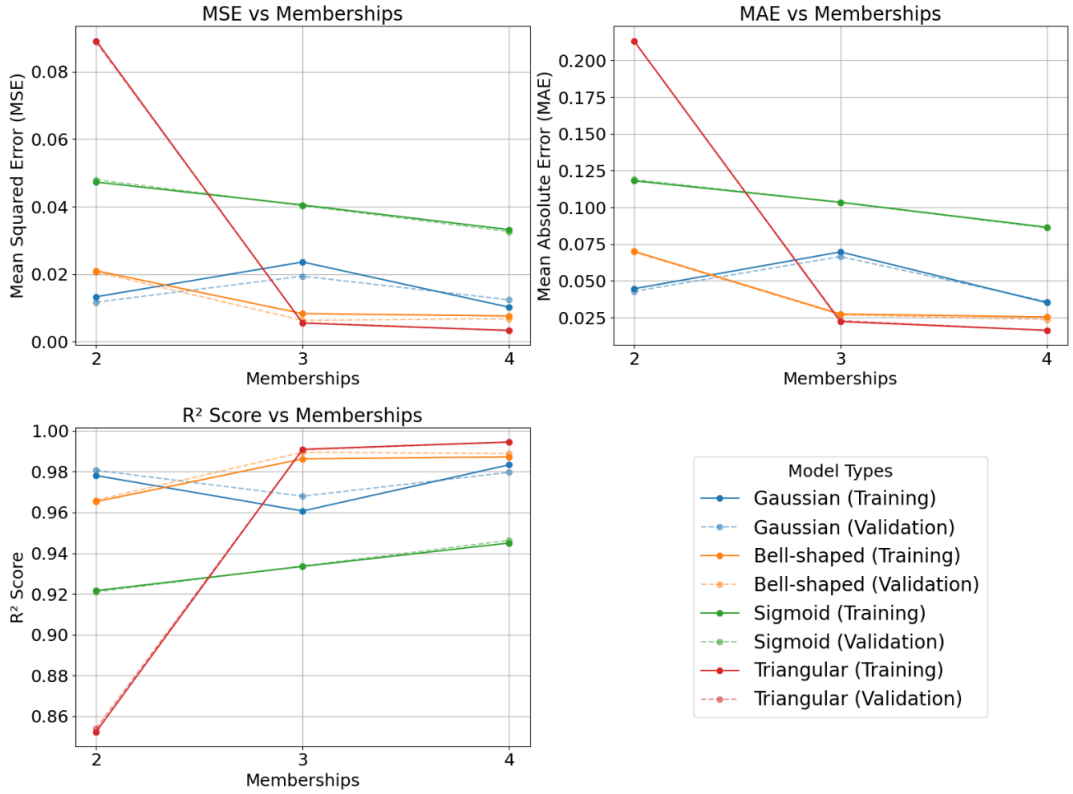

Figure 2-30: Adaptive Neuro Fuzzy Inference System (ANFIS) evaluation results.

The implementation, evaluation, and KPI measurement of the AI-enabled RT zero-touch control loop analysis function is still in progress in the context of the project PoC#C, and further results will be provided in the

context of deliverable D2.6 in WP2. These include the measurement of service latency at runtime, the time required to instantiate and configure the closed loop function during the service provisioning phase, the time required to complete a full AI-enabled closed loop cycle, demonstrate the ability of integrated intelligence in the analysis stage, and the reduction of human interventions over the managed entities.

2.2.1.2 Penalty-based management of concurrent service Control Loops

This solution is included in the "RT zero-touch control loops automation & coordination" functionality of the management framework. The main goal is the management of simultaneous closed loops running different services each of them with specific KPIs to be fulfilled. The implementation targets the service assurance as it is managing different closed loops with the goal of fulfil the expected KPIs of each of them. Furthermore, as different actions are proposed (e.g., to increase the computation capability of a specific node of the network to support a specific closed loop) the solution also aims to provide the network resource configuration functionality.

The implementation addresses the challenge of managing distinct closed loops running different services, each with unique KPIs to be fulfilled. The challenge in managing multiple closed-loops arises from the fact that specific actions executed by one closed-loop may improve its performance while simultaneously deteriorating the performance of another concurrently running closed-loop. Therefore, before a closed-loop decides to execute a particular action, the CL action proposal must be sent to the closed loop coordination service capable of identifying potential conflicts with other closed-loops. Therefore, to manage such conflicts, the "Intent Management Function" (IMF) component is introduced. The intent concept here is the same as the one provided in WP2 [HEX224-D23] and a closed-loop is created to fulfil the KPIs provided by the stakeholder. For example, a stakeholder can interact with the DSP architecture provided in WP2 to create an intent for an Ultra Reliable Low Latency Communication (URLLC) service with a minimal latency requirement. A closed loop is created to monitor the URLLC service and provide the necessary actions to meet the service latency KPI. Therefore, the focus in this Section is in the closed-loop management and conflict resolution between simultaneous closed-loops.

The IMF operates through different stages in its task: monitoring, which involves collecting all the metrics necessary to evaluate the condition of each closed-loop; the proposal stage, used to propose actions aimed at fulfilling KPIs; the prediction stage, which forecasts the future state of the action; the evaluation stage, which analyses and decides on the optimal action for a specific closed-loop; and the actuation stage, where the chosen action is executed and its impact is monitored. It is important to note that any action (e.g., allocating additional computational resources to a closed-loop) is designed to minimise damage across all closed-loops. In other words, the chosen action considers the overall system impact rather than focusing on a single closed loop. With this global perspective, the IMF can make optimal decisions and reduce the impact of conflicts across different closed loops. If an action is taken solely considering one closed-loop without accounting for the overall impact, the proposed action may benefit the targeted closed-loop but could deteriorate the performance of another independently running closed-loop. The workflow in Figure 2-31 provides further details on the proposed solution. The workflow consists of three main components: a single closed-loop, the closed-loop coordination, and the conflict management. The workflow begins with a CL collecting metrics (Monitoring) and analysing the appropriate action to take (Analysis). Once a decision is made, it is sent to the CL governance. Before executing the action, the governance forwards it to the Closed-Loop Coordination Service, which contacts the Conflict Management component to verify whether the proposed action could lead to potential conflicts with other closed-loops. After this analysis, the result is sent back to the Closed-Loop Coordination Service, which informs the respective CL governance about the recommended action. In summary, this workflow involves a closed loop monitoring metrics and proposing actions to achieve its specific KPIs. However, before executing the proposed action, it is sent to the Conflict Management component to ensure that it will not lead to conflicts with other closed loops. The outcome of this analysis is then sent back to the closed loop, detailing the action to be taken.

To evaluate the overall performance, a penalty metric is introduced. This penalty is a numerical value associated with each closed loop and it is equal to zero if the CL can meet the expected KPIs. Otherwise, a numerical value $p \in \mathbb{R}$ is associated with the corresponding CL. The goal of the implementation is to take the action that minimises the overall penalty

$$P = \sum_{p=0}^{n-1} p_i$$

where p_i corresponds to the penalty associated with the respective closed loop. In the equation, n is the number of closed loops.

As a numerical example, consider three different CLs named A, B, and C. Each CL is running a different service (e.g., CL A is running a video service, CL B is a mIoT service and CL C is a critical IoT application service) and each of them has requirements that need to be addressed. For example, CL A needs to have a packet loss of at most 1%, CL B needs to have a latency of at most 95 ms, and CL C needs a latency of 50 ms at most. If a specific requirement is not achieved, a penalty value is added. The value of the penalty is previously defined before the CL starts to run. For example, the penalty for CL A can be 10e, for CL B can be 5, and for CL C can be 7.

Consider the scenario in which CL A is running as expected but CL B and C do not operate as expected and the KPIs are not achieved. In this case, the total penalty is 5+7 = 12 (5 for CL B and 7 for CL C). To minimise the latency, some action can be taken. However, these actions need consider the minimisation of the total penalty. For example, if more resources are provided to CL B, than it can affect the performance of CL A.

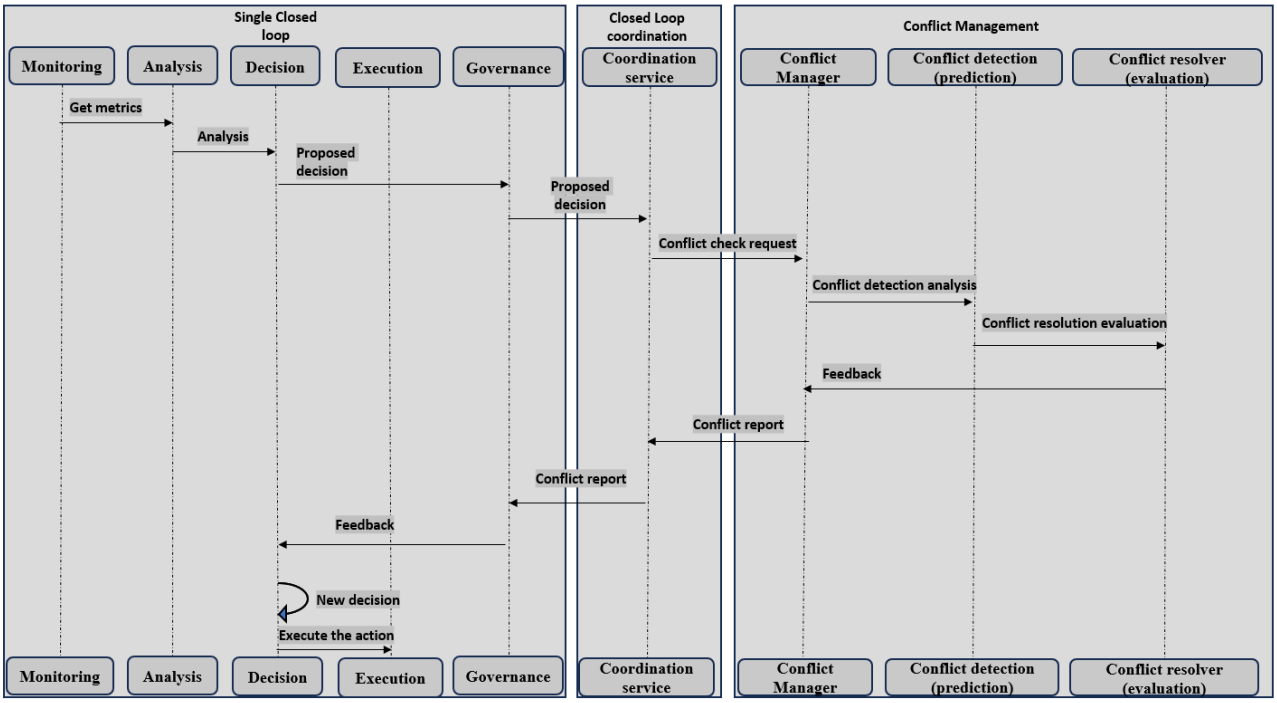

Figure 2-31: Proposed workflow of the closed loop conflict management.

To evaluate this approach, the implementation was conducted using a network emulator capable of emulating varying numbers of User Equipment (UEs), two gNBs (each connected to exactly one UPF), and three domains—edge1, edge2, and central. The primary difference between the edge and the central lies in the computational power. This network emulator supports three services, including conversational video, Ultra-Reliable Low Latency Communication (URLLC), and massive Internet of Things (mIoT). To showcase the effectiveness of the solution, the number of five different closed loops was chosen but the solution can work with a different number of closed loops. Furthermore, each service has KPIs with distinct values that must be achieved. For instance, the conversational video service has KPIs such as a Minimum Bit Rate (MBR) of at least 5 Mbit/s. For the massive Internet of Things (mIoT), two different service instances, referred to as mIoT1 and mIoT2, were created, each with specific KPI values: latency (end-to-end, from UE to the central domain) was 170 ms for mIoT1 and 165 ms for mIoT2, packet loss not exceeding 1% for both, and an MBR of 10 Mbit/s for each service. For services classified as URLLC, two instances were also created with the following

KPI values: 100 ms latency for LS1 and 95 ms for LS2, no more than 1% packet loss for both, and an MBR of 10 Mbit/s for both.

Each of the five services described above is deployed as an independent closed loop. To evaluate the performance of the IMF in managing all closed loops, a metric termed "penalty" was introduced. The concept behind this metric is that the higher the penalty value, the more issues each closed loop is encountering, and the more their KPIs are deteriorating. A penalty of zero indicates no conflicts, and that all closed loops are operating smoothly. In the implementation, the experiment was executed 22 times, and the average penalty was calculated over time. Figure 2-32 shows the results obtained, with the x-axis representing the experiment execution time and the y-axis represents the penalty value. The overall penalty is obtained by the sum of the penalties of each service. It is possible to see in Figure 2-32 that in the beginning of the experiment, the value of the penalty increases until it reaches the value closer to 55. This pattern reflects the fact that the closed loops are being deploying at different time stamps and the system is starting to take the decisions to minimise the overall penalty. After that, the penalty tends to decrease until reaches a stable value around 10.

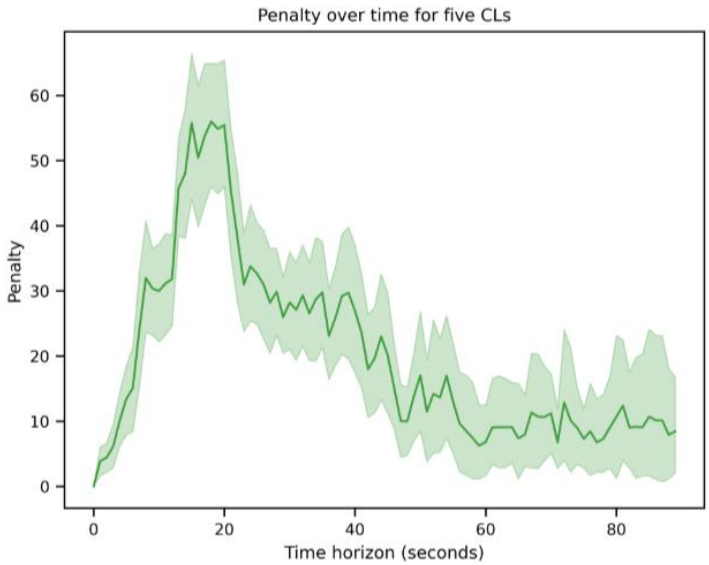

Figure 2-32: Penalty results for the scenario with five different closed loops.

2.2.1.3 Conflict detection for the reactive activities requested by closed loops

The Closed Loop Conflict Detection implementation described here is intended to be part of the Real-time Zero-touch CLs Automation and Coordination functionality. It is built around the Closed Loop Coordination Service, which receives one or more action plans from interacting closed loops. Each action plan consists of a set of activities, with the goal of maintaining the system's KPIs and overall goals. The Conflict Management component is responsible for identifying potential conflicts between these action plans, ensuring that the best combination of action plans is selected for execution.

The conflict detection logic (Figure 2-33) is intended to check if any activities within the action plans act on the same resources. If no conflicts are found, the action plan is executed "as-is". Else, if equivalent actions are found (i.e., activities that are duplicated), they are removed to avoid redundancy. In the case of conflicting activities (activities that cannot coexist), the Conflict Management component decides which action plan should proceed, either partially or entirely, based on predefined policies such as priority, cost, or penalties. The logic also considers if the activity to be disabled is mandatory for the action plan, of it has any dependency with other activities. In those cases, the activity cannot be disabled, and the entire action plan is discarded.

The solution is exposed via an OpenAPI built with Python and FastAPI [FAPI]. The Open API (see Conflict Detection for CLs Automation and Management API in Section 2.2.3.2) allows clients to submit action plans, retrieve conflicts, and fetch the status of running or pending action plans. Key endpoints include submitting new action plans, detecting conflicts, and resolving those conflicts in real-time. SQLAlchemy [SQLA] is used for database interaction to store and retrieve action plans, resources, and conflict data. The overall architecture ensures a dynamic and scalable system capable of handling resource conflicts in real-time. Figure 2-34, shows the functional sequence diagram of the implementation.

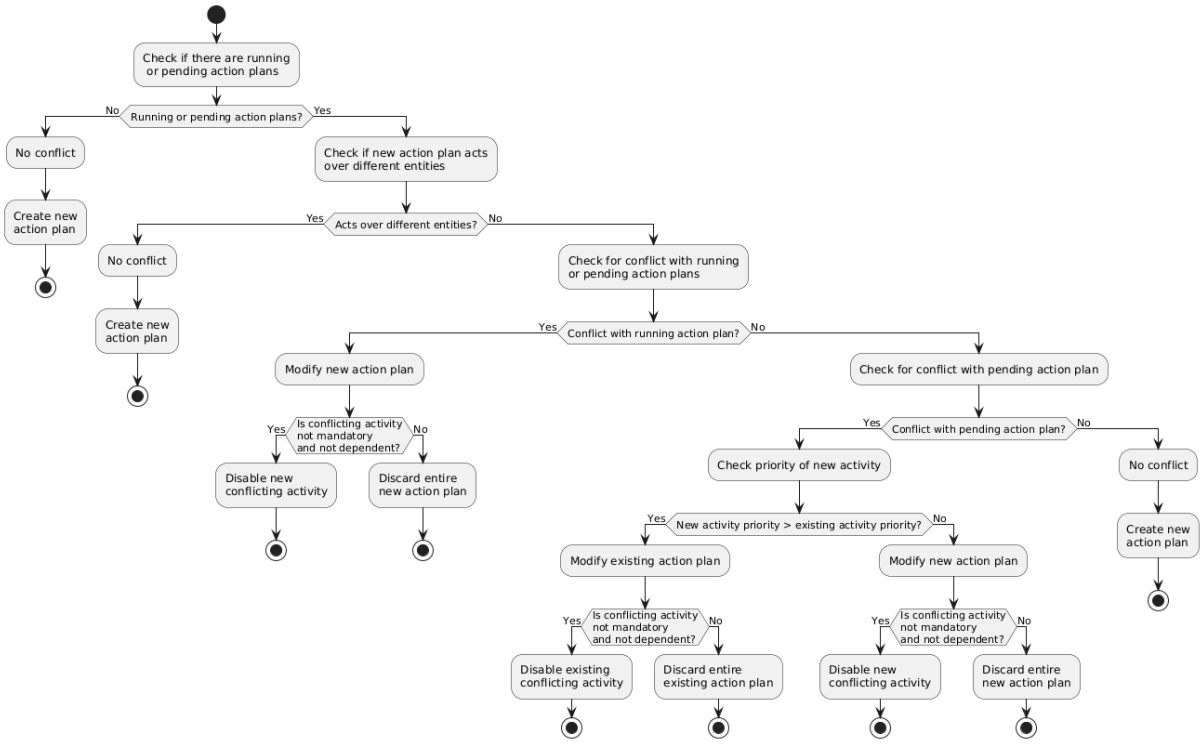

Figure 2-33: Conflict Detection Logic.

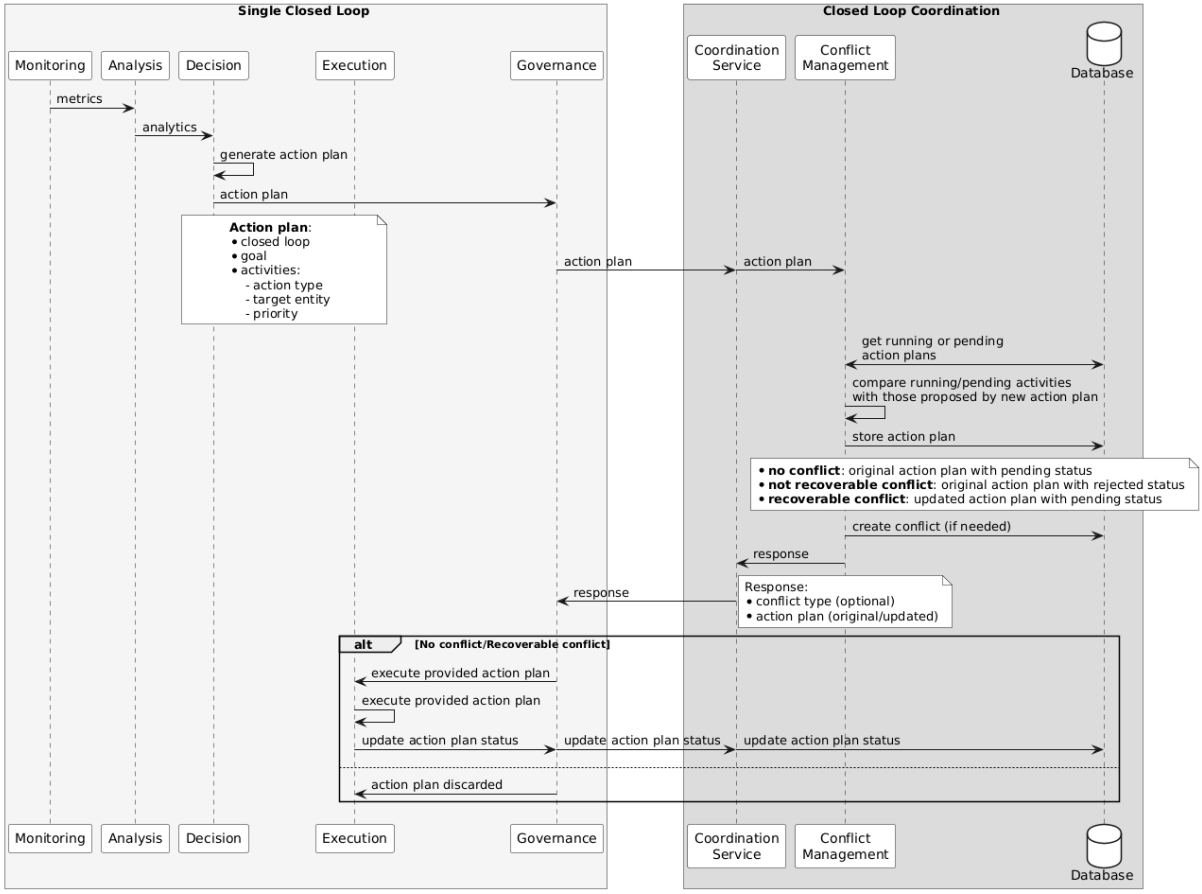

Figure 2-34: Conflict detection sequence diagram.

This implementation considers three levels of priority to select/disable the conflicting activities in the action plans: high, medium and low. These priorities could be defined based on user profiles (silver/gold/platinum) and the committed SLAs, the final use of the managed entities (commercial use, security and rescue forces, etc) or the costs (economic, energy consumption, number of resources used, etc).

The solution is exposed via an OpenAPI built with Python and FastAPI [FAPI]. The Open API (see Conflict Detection for CLs Automation and Management API in Section 2.2.3.2) allows clients to submit action plans, retrieve conflicts, and fetch the status of running or pending action plans. Key endpoints include submitting new action plans, detecting conflicts, and resolving those conflicts in real-time. SQLAlchemy [SQLA] is used for database interaction to store and retrieve action plans, resources, and conflict data. The overall architecture ensures a dynamic and scalable system capable of handling resource conflicts in real-time. Figure 2-34, shows the functional sequence diagram of the implementation.

Experiment results

The development was tested using in-house testbed, validating the conflict management component's ability to identify potential conflicts between concurrent action plans, and to ensure the best combination of action plans is selected for execution based on system policies.

For simplicity reasons, the tests were performed with the use of static priorities, but these priorities should be dynamic and reconfigurable (e.g. increasing with time) to avoid the case of an activity that is never executed due to low priority.

One of the most important and, at the same time, the most difficult part is the definition of the ActionPlan data model so that it can reflect all the closed loops requested actions, and allows establishing relationships with activities, entities and conflicts. This complex data model was approached by breaking it down into the key components, using reusable schemas, modelling relationships clearly, and avoiding excessive nesting.

The experiment was tested with the assumption that a "running" action plan activity cannot be disabled, but future versions should include the consideration that not all the activities of an action plan are launched at the same time, and that some of them can still be pending, so that they can be disabled.

Although conflict detection has impact on availability of services and connectivity, as well as in the reduction of operational costs in terms of energy consumption (avoiding the execution of conflicting actions that should be fixed) and maintenance costs, the most important indicator addressed is the reliability**,** as the ability of the system to perform without failure, which was computed as follows:

$$Reliability(\%) = \left(\frac{Number\ of\ correct\ action\ plans}{Number\ of\ total\ action\ plans\ generated}\right) x\ 100$$

where the number of correct action plans is the sum of not conflicting action plans executed and the conflicting action plans that were updated and executed (reliability increases thanks to the detection and update of conflicting action plans).

In the testbed context it was not possible to obtain specific and relevant KPI values since, as the implementation was not integrated into a real system, the Action Plans needed to be generated artificially which does not allow for the extraction of precise information from the KPIs for conflicts and their possible interdependencies . Therefore, the tests were focused on validating the detection of conflicts itself, and not on performance. As a valid reference in the state of the art, it has been seen that in the context of the Open Radio Access Networks [DMIC23], at least a 2% increase in mean bitrate was achieved by relying on conflict detection techniques, which it is considered to validate the approach taken here. This conflict detection implementation follows a comparison approach where new proposed action plans are compared with the existing ones. This could be extended with AI/ML-based techniques that can help with the task of identifying conflicts and guarantees the best selection of activities for their execution.

2.2.1.4 Human-assisted training of cognitive closed loops functions for network automation

This implementation is also related to the RT zero-touch closed-loop automation functionality with the aim of introducing more cognitive feature for CL operations. The cognitive feature comes from causal reasoning capability by utilizing causal information in the network. This causal reasoning capability can enable the

network to learn the cause and effect among the network KPIs and configurations by using the logic and facts that can be collected from the network. Usually, causal reasoning is realised through a causal graph [JP09] where there are nodes and edges. Nodes can represent a network KPI, metric, or a configuration parameter whereas edges represent a causal relation between nodes. To achieve the cognitive capability, a causal discovery algorithm has been developed. There are two phases in the discovery: the so-called online and offline phases. In the offline phase, there is an offline data set which is used to find the candidate graphs relying on the causal discovery algorithms [PC91]. In the online phase, real network data is actively collected when different actions are taken and the outcome of these actions is monitored and compared with the candidate causal graphs.

There are two main challenges associated to this approach: first, to discover candidate causal graphs (possibly a complex graph with a high number of nodes) with best accuracy, and second, to find a method to verify and update the candidate graphs to make them more robust to changing environment and networking conditions**.** The state-of-the-art algorithms [PC91] usually take observational data and apply specific methods such as conditional independence tests and aims at achieving a final graph. However, it may not always be possible to monitor data corresponding to all different environments and conditions, which is key for the enhancement of causal discovery. The developed idea was implemented on an in-house network emulator working with real traffic (i.e., real video traffic in the emulator). The target was to apply this causal knowledge algorithm on the Analysis and Decision stages of CLs. In the implementation, a common networking scenario was created with different UEs and services, such as video and URLLC, and defining a number of actions and KPIs associated to different services. The causal relation among the actions and KPIs was investigated with the developed idea.

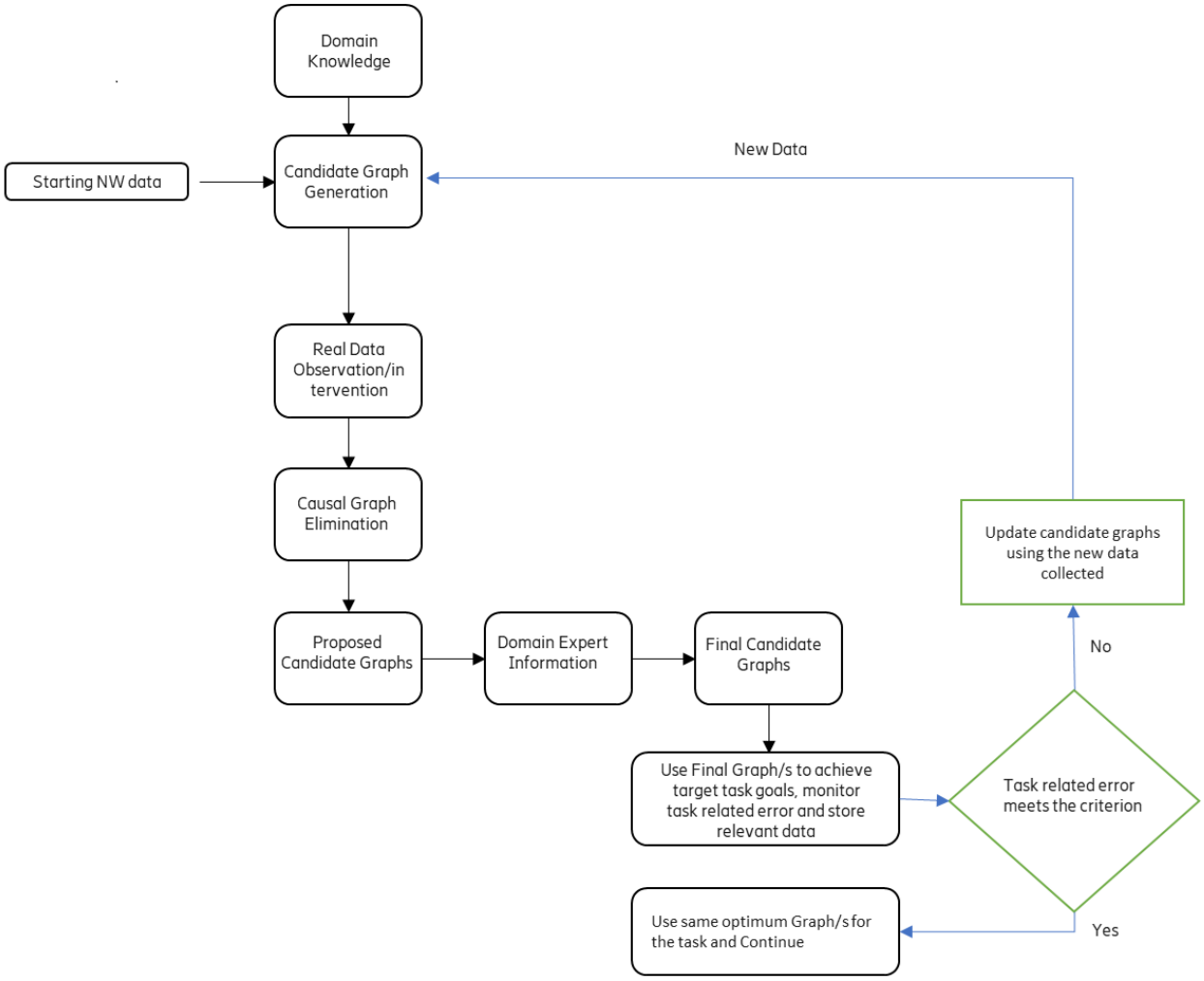

Figure 2-35: Continuous Causal Discovery Flow.

Figure 2-35 shows the causal graph discovery flow. As it can be appreciated, as a first step, possible candidate causal graphs are generated relying on the available data. However, only few of those graphs can represent the

true data generation in the context the data is generated by the actual causal factors. The second step is that, as the number of candidate graphs can be large, a domain experts' team further eliminate some of these candidate graphs, so finally only a few candidate graphs are selected as final graphs. Figure 2-36 represents the sequence diagram associated to this implementation. Although a human expert can get involved during the process of causal graph generation, it must be noted that the graph generation is done before a RT zero-touch CL starts its operation. In other words, the generation of a casual graph does not impact the real-time property of the CL, and that the CL can utilise the causal information for its operations in real-time.

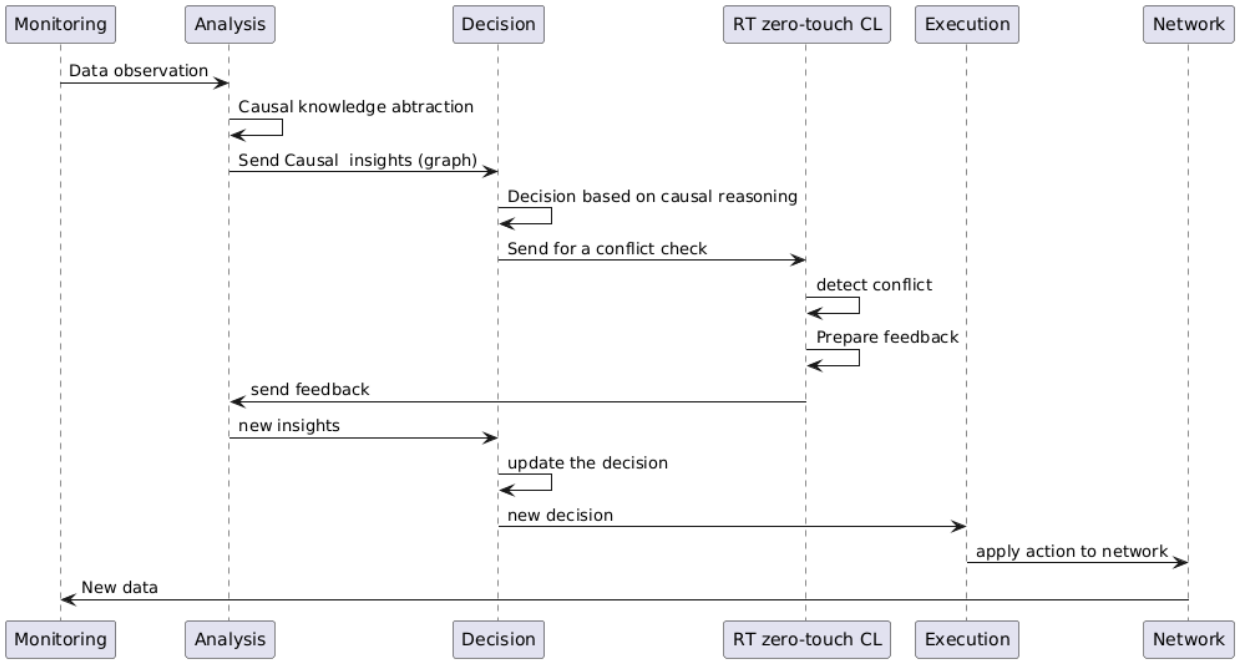

Figure 2-36: Sequence Diagram of Utilizing Cognitive CL.

The Cognitive CLs are fed with the Monitoring and Telemetry data. Then, a data-driven approach is applied to discover the best causal graph which, once achieved, is propagated to the Analysis and Decision stages of the CLs. Next, these stages utilise this information to make causal reasoning that brings not only a more effective system solution, but also more explainable information behind the choice of different network configurations compared to a system without cognitive and causal capability.

Experiment Results

In order to validate the ideas towards integrating causal learning into cognitive operations, an in-house network emulator was used in a lab experiment. The network emulator consisted of containerised entities including network functions, UEs, and application instances. These individual components were implemented as web servers providing a set of APIs enabling configuration management. Based on these capabilities, the set of actions that can be executed to reconfigure the network is as follows:

I. Parameter 1: MBR (Maximum Bit Rate); Changing the maximum throughput that can be achieved by a particular UE. This is actually a network configuration.
II. Parameter 2: Propagation delay; It is a system parameter within the emulator. The propagation delay can be changed by adjusting this parameter. Consequently, it also changes the round-trip time (RTT) between the video service and the UE.

Conversational video is implemented in the network emulator, representing an Enhanced Mobile Broadband (eMBB) type of service that is latency intolerant. The relevant KPIs measured in this case were QoE, throughput (TH), and packets loss (PL), while the control parameters were MBR and RTT. The developed algorithm was applied to capture the causal relations between these KPIs and the control parameters. First, four candidate graphs were generated. Then, the online phase of the algorithm previously explained was applied (Figure 2-35). Based on the monitored data, the mean square error of QoE was calculated by using the learnt function for each node, relying on the causal parents in each candidate graph. Finally, the candidate

graph with the minimum MSE was selected as the best graph. The minimum MSE was achieved with the graph in Figure 2-37.

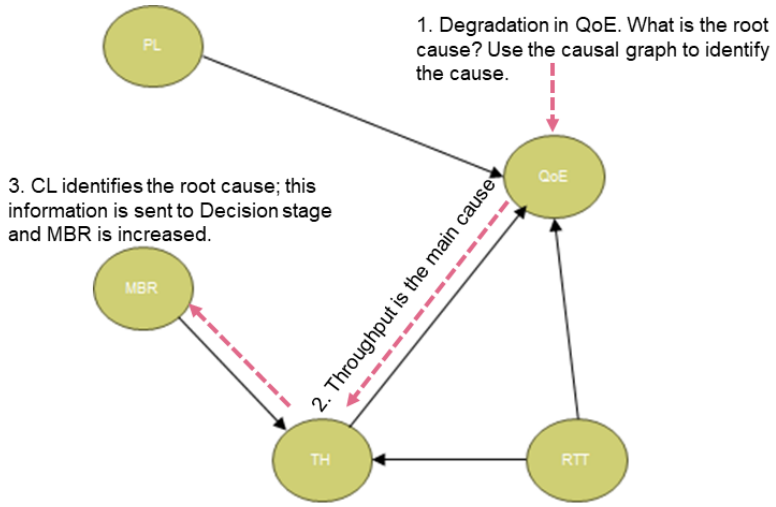

Figure 2-37: Final Causal Graph in the Cognitive CL Experiment.

Here, more than one candidate graph can also be proposed. The proposed final graph was considered aligned with the domain knowledge within an acceptable level. From the graph, one can conclude that the throughput (measured in the range from 0 to 3 Mbit/s) was causally impacted by the MBR (in the range from 0 to3 Mbit/s) and also by the RTT (in the range from 0 to 200 ms). Also, the PL (between 0 to 1), TH , and RTT were the main causal parent of the QoE (in the range from 0 to 4). This final graph could be used by a CL to optimise its task. For example, one could assign a CL to achieve a target related to QoE, and the CL would utilise this final graph to find out the best possible network configuration given the context at the decision time. Annex A.3 gives more information on causal graph and its usage for CL operation.

2.2.1.5 Sustainable MLOps implementation

This implementation targets the Sustainable Machine Learning Operations (S-MLOps) component of the smart management framework, which aims to provide stakeholders with a solution for the automation, monitoring, and optimisation of workflows involved in the continuous development, deployment, and operation of AI/ML-based network services within a multistakeholder scenario. In this context, a workflow is defined as a structured sequence of tasks designed to automate, manage, and streamline the lifecycle of AI/ML-based network services, encompassing data preparation, model development, deployment, monitoring, and continuous optimisation. Furthermore, the S-MLOps component places a strong emphasis on measuring energy consumption across the stages of these workflows, reflecting its commitment to sustainability and operational efficiency.

The implementation consists of two parts:

a) A simple command-line interface (CLI), enabling different stakeholders (e.g., Software Vendors, network operators, etc.) to connect to their respective domains and deploy AI/ML-based software artifacts to collectively form their customised MLOps workflows. This CLI facilitates the management and orchestration of these workflows, ensuring a structured approach to the deployment process. Figure 2-38 shows a screenshot of the S-MLOps CLI.
b) The software components to collect the sustainability related data to be added as metainformation to the different MLOps workflow stages, as well as the associated GUI.

The implemented CLI allows stakeholders to construct workflows tailored to their specific domains while also enabling the retrieval and sharing of information with other stakeholders, such as datasets, models, and related artifacts. This capability facilitates the creation of a multi-stakeholder workflow, composed of the individual workflows developed by each stakeholder, fostering collaboration and integration across domains.

Through the CLI, stakeholders can easily connect to their respective domains and deploy a set of predefined modules by executing individual commands, with each command corresponding to a specific module to be deployed. This approach allows for the seamless creation of workflows customised to the needs of specific use cases, such as workflows focused on data preprocessing, ML model development, or the deployment and validation of trained models in a production environment.

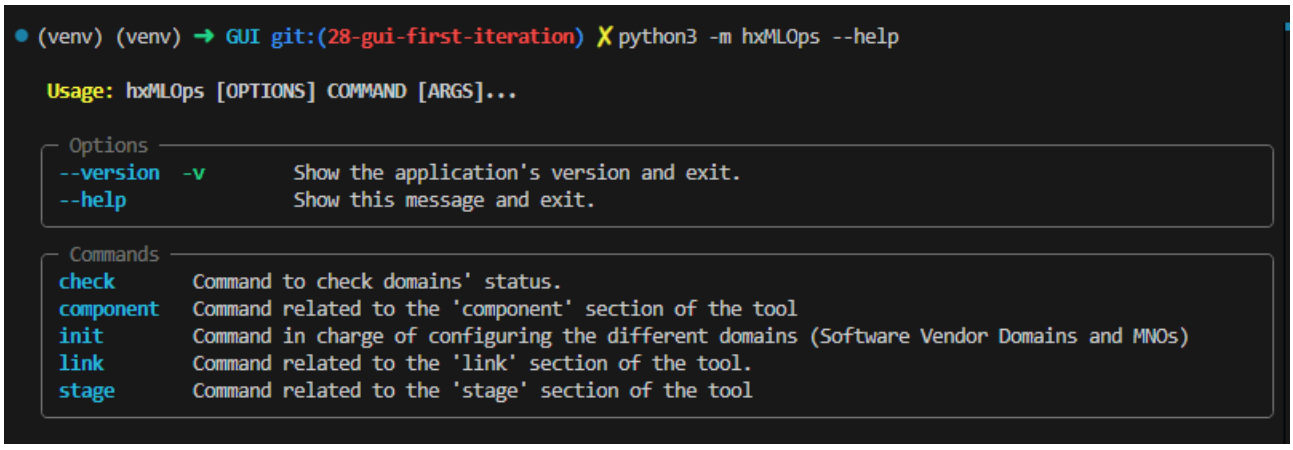

Figure 2-38: Screenshot of the S-MLOps CLI.

The CLI supports the deployment of multiple modules which are categorised, based on their functionality, into the following three groups: **Components**, **Stages** and **Links**. The different modules are identified in Figure 2-39**,** and their purpose is explained below:

- **Components**: It refers to state-of-the-art frameworks, predominantly open-source, that are designed to address specific functionalities within a particular domain. These frameworks play a crucial role in enhancing the efficiency and scalability of MLOps workflows. The components integrated in the CLI are the following:
  - **Storage components,** responsible for managing data storage within the workflows. Specifically, databases such as TimescaleDB [TDB] and PostgreSQL [PSQL], as well as MinIO [MIN] and time series storage like Prometheus [Prometheus] for sustainability metrics are available for use.
  - **ML toolkits**, comprehensive platforms designed to manage the entire lifecycle of ML/AI models. These toolkits support a range of functions, including data preparation, model training, experimentation, versioning, and deployment. The CLI includes a command for Kubeflow [KBF] deployment.
  - **Serving platforms**, frameworks designed to enable the deployment, management, and operation of ML/AI models in production environments. The CLI provides commands to deploy instances of TorchServe [TS] and TensorFlow Serving [TFS] for serving PyTorch [PTC] and TensorFlow [TF] models, respectively.
  - **Energy measurement asset**: Modules related to energy measurement. The software includes Kepler [KP] and Scaphandre [SPD]**,** with plans to further expand this catalogue in the near future, as well as sustainability agents to add information about carbon intensity depending on stakeholder location and services to group metrics by MLOps workflow instances and stages, allowing the delivery of sustainability information as part of process characterisation.
  - **Observability**: Including Grafana [GFN] as visualisation tool for observability and visual representation of sustainability metrics evolution in a multistakeholder scenario.
- **Stages**: It encompasses all modules that provide specific functionalities within the lifecycle of designing a ML/AI model. The tool includes modules for **data encryption and anonymisation**, which are particularly useful for preserving data privacy in a multi-stakeholder environment, as well as data validation and model training and testing. In the future, the catalogue is planned to expand with specialised modules related to the develop of network services based on AI/ML models.
- **Links:** It includes all modules that facilitate the exchange of information, such as datasets or trained models, across different domains (e.g., software vendors, network operators). Additionally, these

modules enable the transmission and retrieval of information to and from modules created within the component and stage sections. The CLI includes two Open APIs. The first, named **Model Sharing**, functions as a model registry, enabling the storage of trained ML models for subsequent sharing across different domains. Additionally, a second Open API, named **Dataset Sharing**, has been developed to facilitate the historisation and sharing of datasets specifically designed for trained ML/AI models.

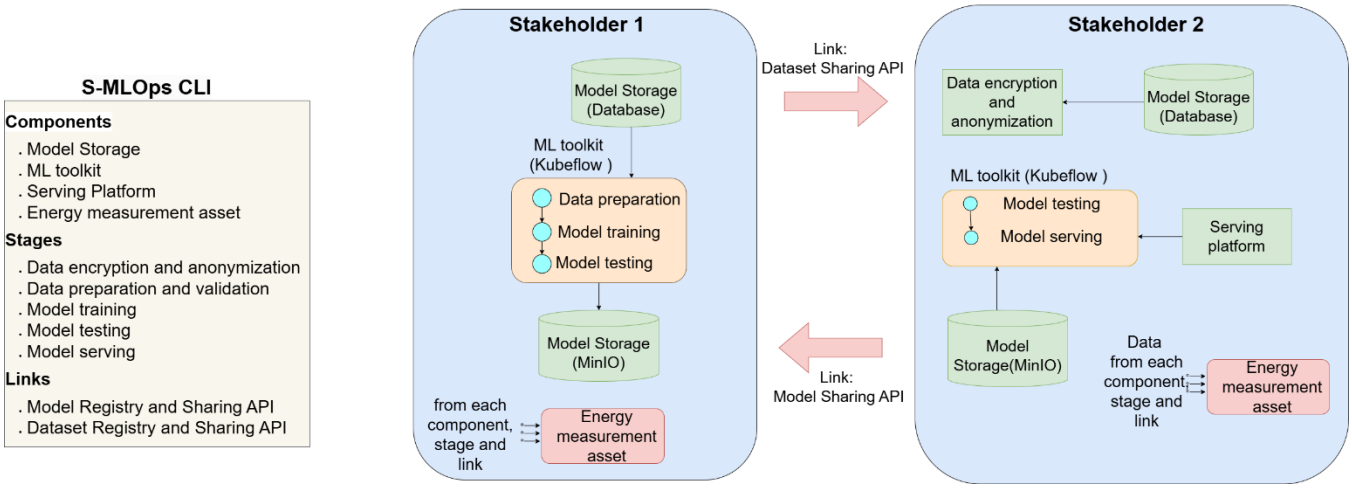

Figure 2-39: Visual representation of the types of modules included in the S-MLOps CLI.

Evaluation

To evaluate the capabilities offered by S-MLOps, a test environment has been set up to simulate a telecommunications scenario featuring a software vendor and a mobile network operator. Each domain is composed of a single-node Kubernetes cluster configured as an Amazon Web Services instance. Using the S-MLOps CLI, the clusters were accessed to configure an MLOps workflow in each of them. These workflows are independent but interconnected through the "links" provided by the CLI, resulting in the creation of the overall multi-stakeholder Sustainable MLOps workflow described in the Figure 2-40.

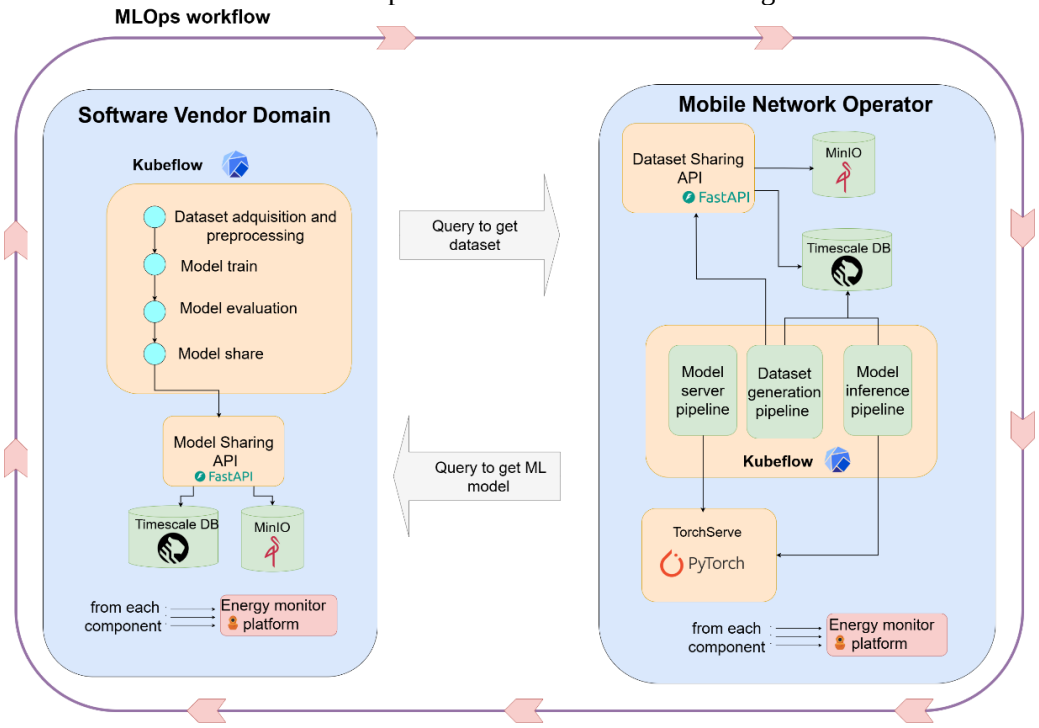

Figure 2-40: Illustration of the Sustainable MLOps workflow for testing purposes.

On the Software Vendor side, a workflow focused on the training and validation of AI/ML-based network services has been configured using components, stages and links from the S-MLOps CLI. TimescaleDB and MinIO were deployed for storing datasets and saving the developed models, respectively. The Kubeflow

framework was used to manage and deploy workflow stages, including stages for data preparation, where datasets were retrieved from the MNO domain, as well as model training and evaluation stages. Additionally, the Kepler framework measured the energy consumption of components, stages, and links, and the Model Sharing API was deployed to store trained models and facilitate their sharing with the MNO.

On the other hand, the network operator domain has been configured to perform two main tasks. The first one involves the creation of datasets to be shared with the software vendor. To accomplish this, a TimescaleDB database was deployed to simulate the production data, along with the Dataset Sharing API to register and share dataset versions. The second task focuses on the deployment of models for subsequent inference. For this purpose, the Torch Serve module was deployed. The logic for model deployment, dataset creation, and the inference stage has been implemented within internal pipelines orchestrated using Kubeflow.
During the execution of the MLOps workflow, at the same time the AI model is trained, evaluated and shared as well as when execution is ongoing, energy consumption measurements and carbon intensity translations are being performed (example in Figure 2-41), considering the impact of the different stages in the workflow in the MLOps environment described (storage, toolkit, frameworks), and identifying by run and task the sustainable metrics.

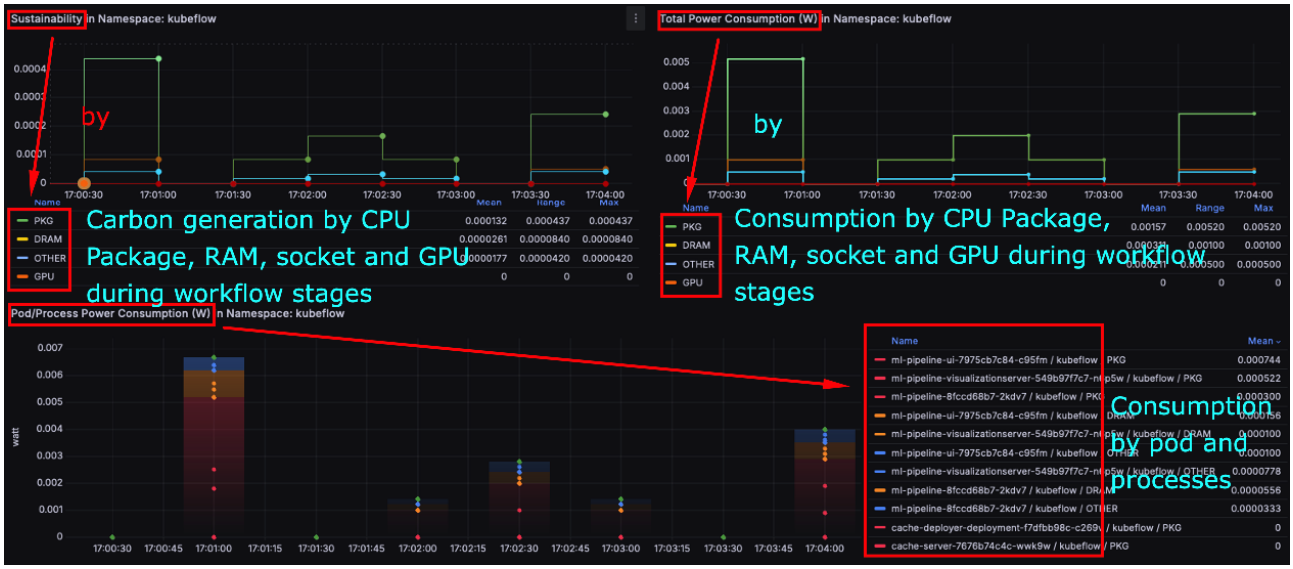

Figure 2-41: Workflow and energy consumption measurement illustration.

These metrics provide the capability to add specific metainformation to be shared, as part of the MLOps workflow in a multistakeholder scenario, as characteristics to identify the cost in terms of sustainability of model training, considering data enrichment and preprocessing, testing, evaluation and/or execution, as well as traditional metainformation regarding model sensibility, accuracy and performance, given the MNOs to select and use models depending on scenarios and convenience (temporal slots, locations, core situations). These information and models, as well as catalogue, are shared through APIs designed following OpenAPI specifications (see Section 2.2.3.2 OpenAPIs, Table 2-2, Sustainable MLOps models sharing API).

The experiments performed started with the dataset creation pipeline to gather the data to be shared for training and model generation purposes as shown in Figure 2-42. In this stage all the processes related with the task in the pipeline were monitored to generate sustainability metrics by process, and also, aggregated by task and pipeline.

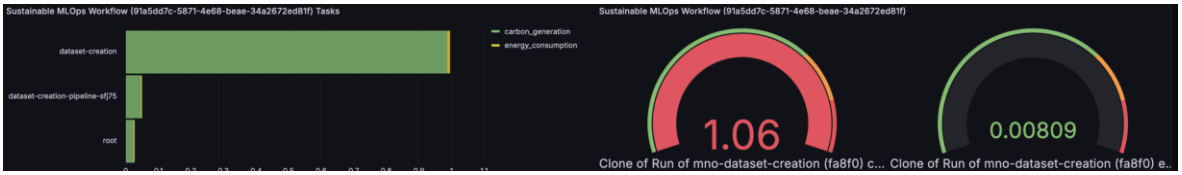

Figure 2-42: Data generation stage with sustainability measurements.

These data are exposed by an API (illustrated in Figure 2-43 and listed in Table 2-2, Sustainable MLOps Workflow Info Collector API) to be consumed, aggregated and exposed by the model sharing API as model

sustainable characteristics to be used by MNO, for instance, to select models and versions by performance as well as by efficiency.

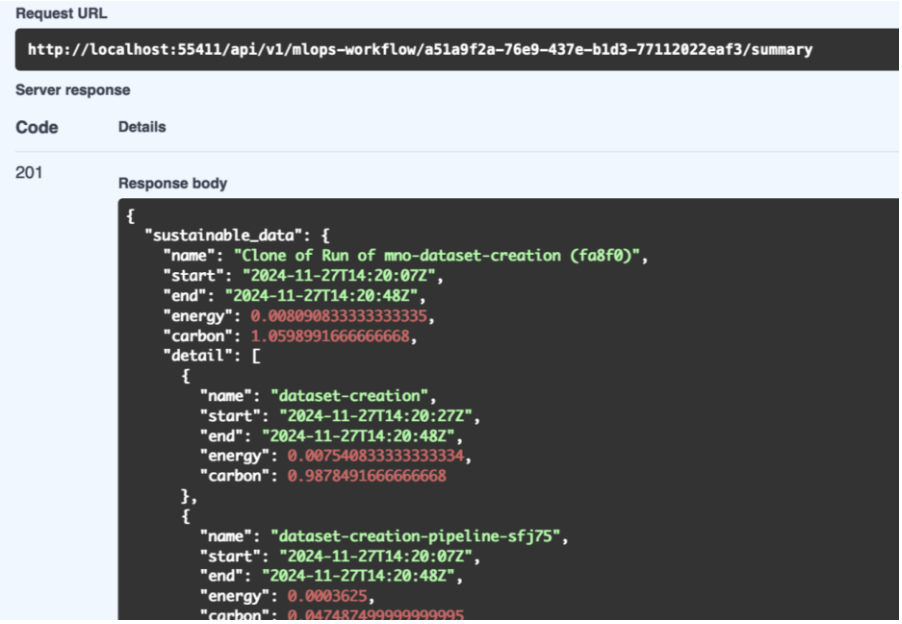

Figure 2-43: Sustainable API example for Kubeflow Pipelines.

The next stage of the workflow is the AI model training, where a pipeline is defined to process the generated dataset, to train a model with specific hyperparameters, and to evaluate it to deliver statistical information related with its efficiency, as well as sustainable data from the monitoring and metric generation agents deployed at the software vendor environment), as shown in Figure 2-44.

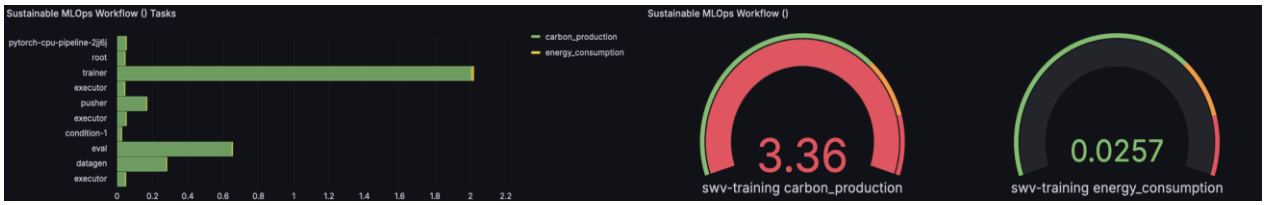

Figure 2-44: Training and Evaluation stages with sustainability measurements illustration.

Once the model is trained and evaluation and sustainability data is retrieved and added as metainformation, the model is shared through the Model Sharing API. After this, the next stage is the MNO model acquisition from the software vendor (once the selection of the version based on the model parameters and characterisation was done), as shown in Figure 2-45.

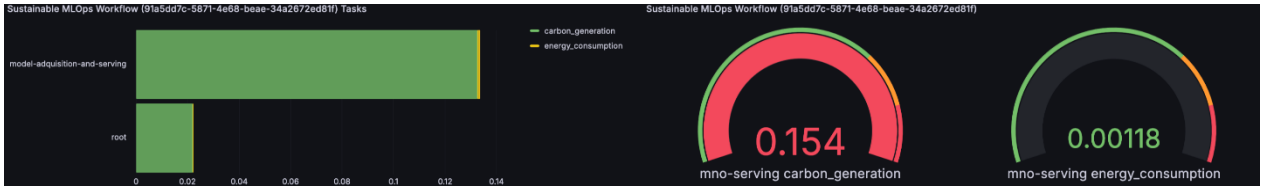

Figure 2-45: Model acquisition with sustainability measurements illustration.

Finally, the acquired model execution, where the same measurements were monitored to extract sustainability related metrics to be added to the AI model metainformation, and to check and validate the software vendor sustainability indicators for the model version in use, as shown in Figure 2-46.

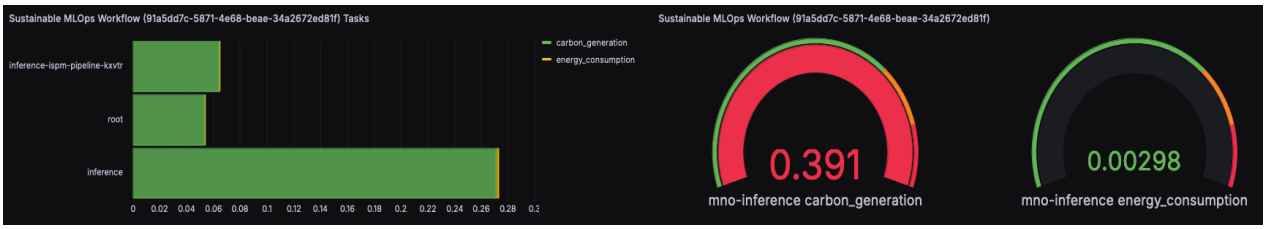

Figure 2-46: Model inference with sustainability measurements illustration.

The above configuration demonstrates a high level of integration and automation, leveraging tools such as Kubeflow, TimescaleDB, MinIO, and dedicated APIs for model and dataset exchange, thereby optimizing workflows and minimizing manual intervention. A total of 7 tasks have been automated, with only 2 requiring manual intervention —specifically, the creation of datasets and its download once ready—, resulting in a 77.7% value for the Automation Level KPI. Furthermore, the Efficiency of Automated Decisions KPI has been calculated based on multiple executions of the process in a controlled test environment, all of which were successful. This results in the Efficiency of Automated Decisions KPI having a value of 100%, highlighting the reliability of the automated decisions. In addition, the Energy Monitoring Capability Index (EMCI) KPI has been calculated by evaluating the system's ability to measure energy usage in terms of granularity, coverage, real-time monitoring, measurement accuracy, and historical reporting capabilities. Based on these criteria, the system achieved an EMCI of 80%, reflecting its capacity to effectively track and monitor energy consumption. This tool plays a critical role in identifying energy inefficiencies and provides a foundation for future optimisations aimed at improving energy efficiency. However, it is important to highlight that these tests were conducted in a controlled environment, and further validation is required in a real-world production setting to confirm the system's performance under operational conditions.

2.2.1.6 ETSI TeraFlowSDN related contributions

The implementation of the Programmability Framework of the smart network management network has been performed relying on TeraFlowSDN (TFS) [TFS24] using data-plane in-a-box is a novel feature that allows quick demonstration of TFS and its integration as a transport SDN controller with an emulated transport network that can be easily integrated in a system proof-of-concept. It has been integrated already in the System PoC#B.1 and will be part of PoC#C. Different implementations have been made based on TFS, which are described below.

The first one is the so-called SmartNIC Transceiver support using OpenConfig extensions. This implementation has been presented at [VVG+24]. It has been deployed on the ADRENALINE Testbed [MNC+17] and is planned to be part of future the TeraFlowSDN release 5. Anomaly detection and mitigation is one of the required novel security features for an SDN controller. To this end, this section presents a proposal for implementation of an SDN-enabled anomaly detection mechanism by extending the SDN controller with support for SmartNICs.

Another implementation is regarding Time Sensitive Networking (TSN) and Deterministic Networking (DetNet) to provide bounded latency and zero packet loss in a reliable fashion in respectively L2 and L3 networks. This implementation proposes an East-Westbound (EW) control architecture that allows a modular DetNet network using a divide-and-conquer approach. Multiple centralised network controllers (CNCs) are employed, where each controller is only concerned about its specific network segment. This segment is a collection of devices that employ the same packet scheduling techniques, e.g., a TSN network segment with Time Aware Shaping (TAS)-based switches or an L3 network segment with routers employing strict priority queueing. The controllers use the EW protocol to exchange various types of information.

Another implementation regarding automated transport network re-configuration support is planned to be introduced in TFS release 4 and extended in release 5. This implementation is also planned to be demonstrated in ADRENALINE Testbed [MNC+17].

Another additional implementation is regarding the synergy between the MEC BandWidth Management (BWM) service and TeraFlowSDN (TFS) in dedicating resources for optimal network resource allocation in the gaming domain. This implementation is part of the ETSI MEC PoC 14: Network resource allocation [MECP014] for application specific requests using MEC BandWidth Management service and TFS. The

source code has been introduced in Release 3. BWM empowers applications to earmark specific bandwidth quotas (among other quality of service constraints, such as latency) for gaming applications, while TFS orchestrates the management and control of traffic flows. The resulting prioritisation of gaming traffic over other data holds the potential to significantly enhance the overall gaming experience for users.

Finally, the evaluation and detailed exposition on the intersection and integration of TM Forum, IETF and ETSI, focused on enhancing network management through API standards has been provided in [HEXA223-D63]. The integration of TM Forum APIs with ETSI TFS NBI exemplifies the convergence of distinct strengths from both frameworks to create a holistic, customer-centric solution. From the TM Forum perspective, the integration leverages its standardised Open APIs, such as TMF640 and TMF664, which provide robust service abstraction and orchestration capabilities. These APIs ensure that business applications can interact seamlessly with network services and resources, adhering to widely accepted telecom industry standards. From the ETSI framework, the focus on intent-driven operations aligns with TM Forum's customer-centric approach, emphasizing high-level requirements over technical configurations. Additionally, ETSI's adoption of YANG models for network management serves as the foundation for aligning data models and protocols during the integration process. Together, the technology-agnostic principles of both TM Forum and ETSI facilitate a solution that abstracts the complexity of the underlying network technologies, enables dynamic service and resource management, and supports a unified approach to network automation and flexibility. This synergy ensures that customer needs can be specified independently of the underlying technology, while benefiting from the combined strengths of TM Forum's business-oriented APIs and ETSI's technical and architectural network management frameworks.

2.2.1.7 Monitoring and Telemetry Implementation

Several specific implementations and validation of this functionality have been considered, such as TeraFlowSDN event-driven monitoring, energy monitoring, Monitoring platform for integration in closed loop, Passive/in-band and active telemetry for Time-Sensitive Networks (TSN) and Deterministic networks (DetNet), Data fusion for signals correlation and remediation actions. The following paragraphs summarise them.

TeraFlowSDN release 4 provides a novel implementation following the event and data-driven architectural principles previously detailed. To this end, the new event-monitoring system will be included in part of the System PoC#B.1. New workflow diagrams have been considered to provide an innovative sequence of steps in a data processing workflow where large volumes of data are collected, processed, and analysed to gain insights or trigger automated actions. Each step represents a critical stage in the data lifecycle, from initial extraction to final visualisation.

Monitoring energy consumption and carbon footprint generated through the compute continuum is of high significance in view of 6G. Novel mechanisms have been implemented following the presented architecture for monitoring and telemetry, and they have been validated in the laboratory [MNC+17]. This implementation involves deploying distinct components across the cloud, edge, and extreme edge, all orchestrated under this unified framework.

The presented Monitoring Platform can also be considered in the context of the zero-touch closed loops, so in relation with the RT zero touch CLs automation and coordination system described in Sec. 2.1.2.2. The customisation of the Monitoring Platform is integrated in PoC#B (in particular the PoC with cobots [PBM+24]), where the collection of monitoring data feeds two concurrent and hierarchical CLs that cooperate through a coordination model based on delegation and escalation.

Given the critical nature of TSN and DetNet, monitoring is a very important component to verify that the network can meet its requirements regarding end-to-end latency and packet loss. Thus, different monitoring strategies have been proposed. With passive network measurements, data is gathered by passively listening to network traffic, i.e., without interfering with data traffic. Passive monitoring allows to collect simple statistics at switches and routers, such as bytes sent, lost packets, and other similar statistics. Conversely, active probing is an active monitoring strategy where artificial data packets (probes) are sent into the network, using the same links as the data traffic and collecting statistics along the way. Active probing can be used to e.g., measure the end-to-end delay along a specified path. Since active measurements generate additional network traffic, they interfere with the normal traffic flow, and as such they must be carefully planned. In-band network telemetry

offers low-overhead monitoring possibilities, enabling an end-to-end performance view of the network (including end devices), while not injecting any new packets in the network.

The Monitoring and Telemetry functionality also considers signal data fusion [TAZ+22], with focus on active and passive telemetry of different entities and sources of the system, aiming at facilitating agents to reactively mitigate system and network failures, as well as proactively preventing them altogether by analysing real-time information. A clear methodology of collecting the necessary data from the different entities contributing to a heterogeneous and distributed system is crucial to its efficient maintenance and orchestration. OpenTelemetry [OTL24] is a feasible solution that provides such a methodology for accessing data signals at real-time in distributed architectures. The OTLP (Open Telemetry Protocol) Collector provided by the OpenTelemetry framework comes in two different flavours supporting a variety of distributed data collection designs. The OTLP Gateway can be used for centralised deployments where data collection sources are directly accessible, while the OTLP Agent is available for exporters in distributed locations which forward the collected signals to the OTLP Gateways residing in centralised locations.

2.2.2 Implementations based on the management framework

This section describes some example implementations based on the Smart Management Framework described in Section 2.1, showcasing how this framework can be used in practice. The implementations combine different components of the framework, targeting the following topics:

- The usage of the MCE functionality, considering different use cases: the integration of a vertical industry, and its application to certain failure detection and recovery scenarios (Section 2.2.2.1).
- The M&O of a service on the network continuum, showcasing different M&O scenarios on the network continuum, including the proactive migration of service components, the application of the S-MLOps system, service federation, and the Trust Management functionality (Section 2.2.2.2).
- Functionality allocation in a cobot-powered warehouse inventory system. Showcases the orchestration of a network service providing an automated inventory management solution for warehousing operations using collaborative robots (Section 2.2.2.3).
- ML based configuration recommendation for energy saving, showcasing a fully automated CL-based solution for correcting flow misconfigurations in a scalable way in deterministic networks. The system takes autonomous decision for cells sleep and wake-up (Section 2.2.2.4).
- Resource assignment for federated learning. Shows how telco operators could leverage on rich data sources of connected devices and on the deployment of edge compute resources to provision AI model training as a service, in line with the CaaS paradigm (Section 2.2.2.5).
- Flow Reconfiguration based on Dynamic Monitoring and Closed Loops in Deterministic Networks. Provides an automated CL-based solution for correcting flow misconfigurations in a scalable way in deterministic networks by exploring trade-offs between different forms of telemetry (Section 2.2.2.6).
- Edge convergence over federated resources for the computing continuum. Explores the capabilities of the CAMARA EdgeCloud APIs in the management of the compute resources in the network continuum, and the possibility to extend them to be used with federated resources of external administrative domains (Section 2.2.2.7).

2.2.2.1 Management Capabilities Exposure for Network Service Automation

The integration of diverse components within the smart management framework represents a pivotal element in guaranteeing a unified and automated system. This objective is accomplished by the MCE functionality, which assumes a pivotal function in the administration and exposure of orchestration and management resources. By regulating the manner in which network capabilities are accessed and utilised, the MCE ensures that the system responds in an efficient manner to real-time changes driven by user demands, network conditions, or external requirements. This integration fosters the development of a robust and flexible system capable of continuous adaptive response to the dynamic environment of next-generation networks.

The MCE functions as a central node, facilitating efficient interaction between M&O components. It provides interfaces that expose network capabilities in a modular and scalable manner, thereby supporting dynamic and automated resource management across disparate domains. By integrating these components, the MCE functionality facilitates secure and scalable communication between elements, constructed on a cloud-native

architecture that ensures seamless scalability and integration of new services or network functions. The implementation presented here relies on state of the art technologies such as Apache Kafka [KAF24] and REST APIs for external access, thereby ensuring flexibility and resilience. Apache Kafka implements an event-driven messaging system. It is ideally suited for realizing communication goals in an event-driven architecture, thanks to its high-performance, scalable, and fault-tolerant messaging protocol.

To interact with the MCE, a component must follow three simple steps to establish one or more customised communication channels with other components, as illustrated in Figure 2-47. First, the component interacts with the Security API endpoint to retrieve the necessary credentials for establishing a trusted connection. Once security has been ensured, the component can use the Listing API to discover and communicate with other components that have been onboarded, managing access to its communication channels. The aforementioned component is now in a position to begin the process of onboarding and interacting with other components within the M&O ecosystem, having gathered the information. Ultimately, the component can then be integrated into the M&O ecosystem, enabling it to interact with other components and entities within the system. This streamlined process is crucial for maintaining a secure, flexible, and highly coordinated operational environment.

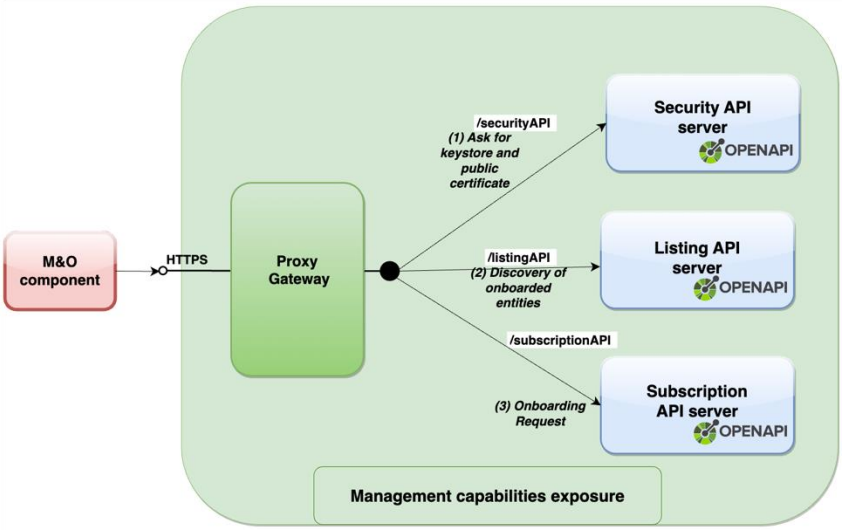

Figure 2-47: M&O framework component first interactions with MCE.

Figure 2-48 shows which M&O components are considered in the description of this implementation based on the M&O framework. One of these components is the Network Programmability system, which enables instantaneous adjustments to the infrastructure network configurations. For this, the programmable network interfaces expose resources such as bandwidth, connectivity, and service provisioning, allowing dynamic reconfiguration in response to evolving conditions, including traffic fluctuations and service demands. This adaptability is considered of particular importance in next generation 6G networks, where real-time responsiveness is vital to meeting constantly shifting requirements.

It is equally important to consider the role of the real-time Monitoring and Telemetry functionality. The monitoring components are designed to continuously collect data from the network, thereby providing a detailed view of performance metrics such as latency, throughput, and resource usage. The telemetry data is then fed directly into the MCE, where it becomes available to other components within the M&O system. The real-time data access enables proactive management decisions. To illustrate, in the event of performance issues, the telemetry data can prompt adjustments to the network configuration via the programmable interfaces. The real-time data flow between monitoring components and programmable network controls establishes a system that can detect and respond to issues as they arise, thereby reducing downtime and optimizing resource allocation automatically.

Another component that will benefit from interacting with the MCE is the real-time zero-touch control loops functionality. Control loops are designed to perform continuous analysis of telemetry data and implement corresponding adjustments to network configurations. The MCE facilitates automation by ensuring that the control loops have access to both real-time data and programmable network interfaces. As changes in network conditions are detected, the control loops autonomously modify network parameters to ensure optimal

performance, obviating the need for manual intervention. This real-time feedback system is essential for maintaining high levels of performance and reliability, especially in complex network environments where manual management would be too slow or inefficient.

Additionally, the Trust Management system plays a crucial role in this implementation by ensuring secure and reliable interactions between different parts of the system, particularly in multi-stakeholder scenarios, leveraging the LoTAF. It interacts with other components in the management framework to maintain the integrity and trustworthiness of the network operations.

Furthermore, the multi-agent system for multi-cluster orchestration is responsible for coordinating actions across multiple network clusters. This system leverages the MCE to facilitate communication and decision-making between agents, enabling efficient management of resources across the network continuum.

Finally, it must be noted that all interactions introduced in this paragraph have been designed but not yet evaluated. These interactions will be evaluated in the context of WP2. The presence of Intent-based solution belonging to WP2 in Figure 2-48, is related to this. Details of this planned evaluation can be found in the evaluation paragraph at the end of this subsection.

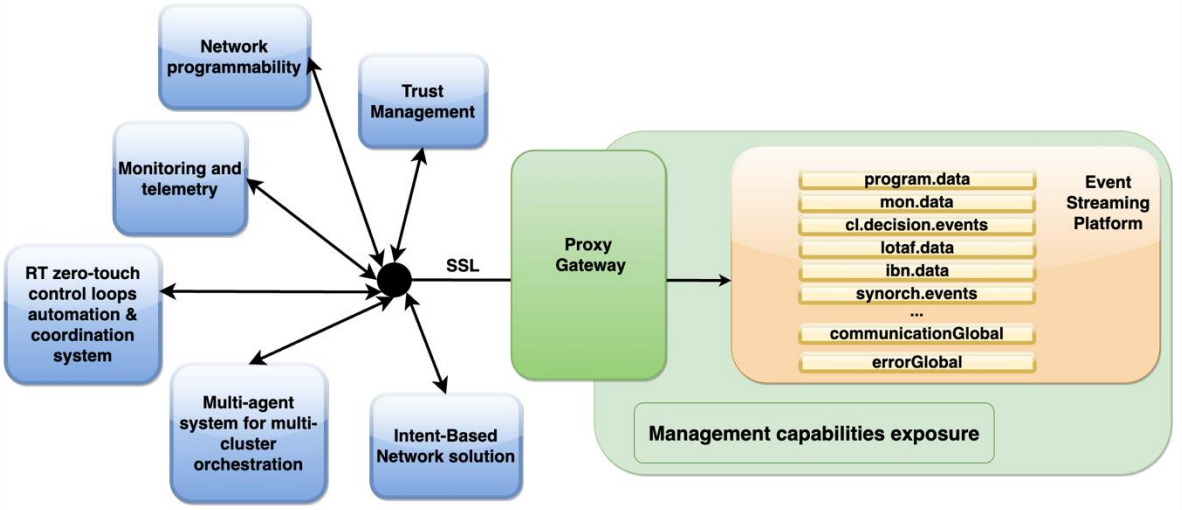

Figure 2-48: M&O framework components event driven interactions overview.

Workflows

This section provides a practical analysis of the functioning of the smart management framework in processes that concern the use of the MCE. The section is divided into 2 parts: The first one describes how the internal interactions within the framework, which rely on the use of the MCE work with the two cases illustrated in Figure 2-49 and Figure 2-50, related to zero-touch CL functions using the MCE service to interact. The former focuses on the registration and subscription phase, which is held during CL instantiation, while the latter focuses on the runtime when the CL functions exchange data and events through the MCE. The second part showcases how a hypothetical external stakeholder could access management capabilities of the smart management framework, and specifically, in the case of the processes described in Figure 2-51 and Figure 2-52.

The first example analysed is a generic interaction, showcasing the onboarding of the RT zero-touch control loops functionality, illustrated in Figure 2-49. During this phase the CL functions are provisioned by the CL Governance and the respective communication channels are created, with topics for each CL instance (e.g., *cl.mon.data* for the data published by the CL monitoring function). The target topics are then configured on each CL function, triggering the subscriptions that enable each CL function to consume its own input. The overall process is coordinated through the CL Governance, which has visibility on the entire set of CL instances and decides the related topics. In detail, the topics used to publish data or events generated by a CL function of type X has the format *"cl.X.data"* or *"cl.X.events"*, depending on the type of content that is to be published. Moreover, an additional global topic *"clg.decision.events"* is created to mediate all the messages from CL Governance to CL Coordination, e.g., to notify the creation of new CL instances or to notify the presence of a CL decision that needs to be validated through CL coordination.

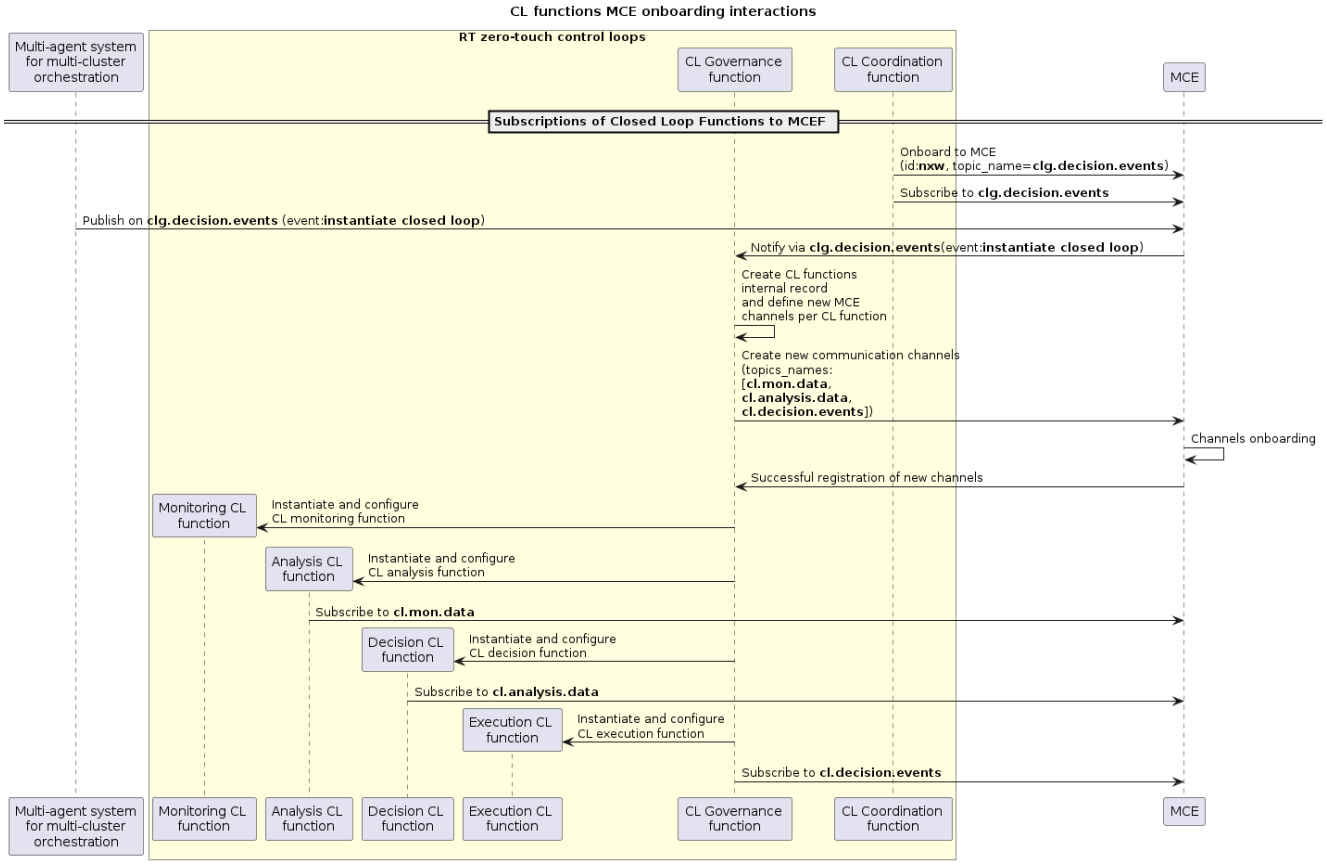

Figure 2-49: CL functions onboarding interactions.

An example of CL functions interaction at runtime (exploiting the communication channels created in the previous step) is shown in Figure 2-50, which refers to the automated migration of video-surveillance processes among robots according to their battery level. It is a well-defined process that ensures continuous operation by enabling the system to detect, analyse, decide, and execute the migration of extreme-edge service components among robots when needed. Various CL functions work in synergy to keep the system performance, migrate operations between robots, and validate these actions through governance mechanisms. Figure 2-50 shows the process that is performed. The steps can be summarised as follow:

1. *Battery Level Monitoring*: The Monitoring CL function continuously collects and publishes battery level data from the robot platform via the MQTT protocol. This monitoring data is then published to the MCE via *cl.mon.data*.
2. *Battery Level Analysis*: The Analysis CL function processes the monitoring data to predict future battery levels, publishing the analysis results back to the MCE via *cl.analysis.data*. The predicted battery data allows the system to anticipate potential failures.
3. *Migration Decision:* The Decision CL function consumes the analysed battery data and other infrastructure details to assess the need for migration. If necessary, the function triggers a decision event, publishing the migration need to move from Robot A to Robot B via *cl.decision.events.*
4. *Governance and Validation:* The CL Governance function validates the decision to migrate by consuming the decision event and further publishing it via *clg.decision.events*. This validation ensures that the migration follows system rules and protocols.
5. *Migration Execution:* The Execution CL function interacts with the service orchestrator to execute the command for robot migration, transferring tasks from Robot A to Robot B once the decision is validated.
6. *Final Migration and Coordination:* The CL Coordination function grants the final permission for the migration after validating the CL decision. Upon successful migration, the system resumes normal operations with Robot B, ensuring service continuity.

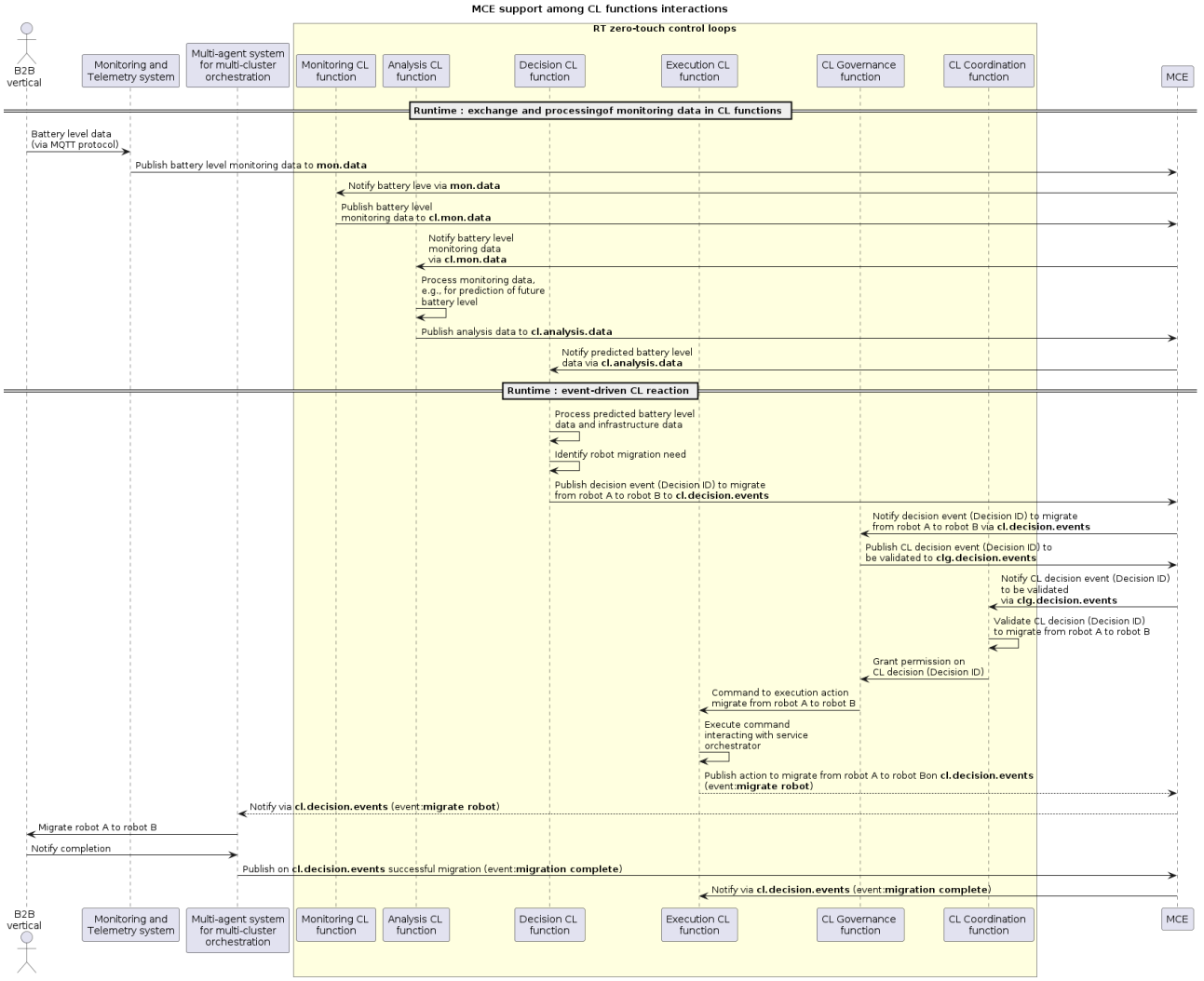

Figure 2-50: CL functions interactions with MCE.

The first use case that shows how an external stakeholder could interact with components of the management framework based on the MCE, is about the recovery of an edge node failure relying on the SDN technology. This is a structured process that enables real-time detection, migration, and recovery when an edge node fails in a software-defined network. In this case, various components must work in harmony to keep the service continuity, reroute traffic, and ensure the stakeholders' trust. Figure 2-51 shows how this use case works. Moreover, in this scenario the Trust Management, thanks to the LoTAF, provides the Level of Trust (LoT), along all the process, to assess the reliability of network components and services. By incorporating these trust metrics, the system can make more informed decisions about fault recovery and service management, enhancing the overall security and dependability of the network. The following outlines the key steps involved in this process:

1. *Edge Node Failure Detection*: The Monitoring and Telemetry Framework continuously monitors the state of edge nodes. Upon detecting a failure, it publishes the event through the MCE.
2. *Deviation in LoT:* The Trust Management System, through LoTAF, consumes the failure event, as the disruption affects the TLA) of the stakeholder. LoTAF adjusts the LoT accordingly to reflect the deviation from the agreed-upon performance.
3. *Service Migration Decision*: The Multi-agent system for multi-cluster orchestration solution reacts to the edge node failure by initiating the service migration process. It publishes a migration request via the MCE to trigger the relocation of services to a new edge node.
4. *Network Reconfiguration*: The Network Programmability system is notified about the edge node failure and the need for traffic rerouting. It reconfigures the network, rerouting traffic from the failed node to the new one, and publishes the successful completion of this task.

5. *Service Migration*: After rerouting, the Multi-agent system for multi-cluster orchestration solution completes the migration of services to the new edge node. Upon successful migration, it publishes the event via the MCE.
6. *Final Monitoring and Service Provider Notification*: The Monitoring and Telemetry functionality resumes monitoring the performance of the new edge node. Once confirmed, the user is notified of the successful recovery and restoration of services.
7. *LoT Recomputation*: After the network returns to its normal operational state and the performance of the new edge node meets agreed thresholds, the Trust Management functionality, thanks to LoTAF, recomputes the Level of Trust to verify that the TLA parameters are fulfilled.

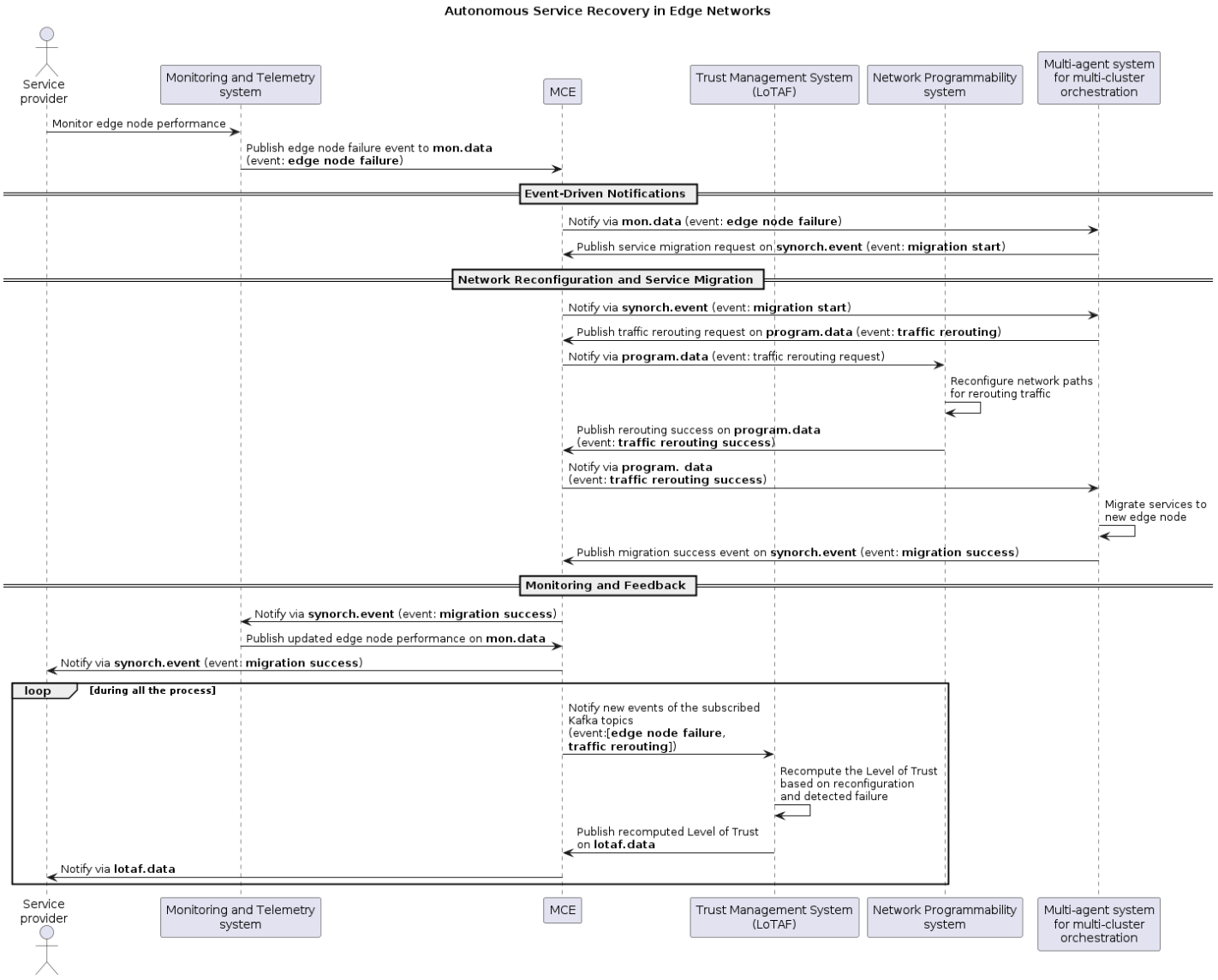

Figure 2-51: Autonomous service recovery in Edge networks use case using MCE

The second use case, considering external interaction, external stakeholder presented is the so-called dynamic fault recovery in SDN, which is a process that ensures real-time detection and automated recovery from network link failures. This recovery is enabled by several integrated components working together to maintain the network stability, reroute traffic, and ensure stakeholder trust. This last step is performed, by the Trust Management system, thanks to the LoTAF, provides the Level of Trust (LoT), along all the process. This process can be seen in Figure 2-52. Below is a summarised step-by-step description of how this process unfolds:

1. *Link Failure Detection*: The Monitoring and Telemetry functionality constantly tracks network health and identifies link failures in real-time. Upon detection, it publishes the failure event via Kafka through the MCE.
2. *Rerouting Request*: Once the link failure event is published, the Network Programmability functionality responds by generating and publishing a rerouting request through the MCE, so ensuring the identification of alternative network paths.
3. *Deviation in Level of Trust (LoT):* The Trust Management system, through LoTAF, monitors events that impact the Trust Level Agreement (TLA) of stakeholders. The link failure and subsequent rerouting affect the level of trust, prompting LoTAF to adjust the LoT accordingly.
4. *Control Loop Reaction*: The Real-Time Zero-Touch Control Loops Automation functionality consumes the rerouting request and initiates the SDN reconfiguration. The system triggers new traffic paths and continuously validates the updated network routes.
5. *Reconfiguration and Validation*: After the SDN controller are successfully reconfigured, the Network Programmability system confirms the success of the operation and publishes the event. The control loop then validates the new network configuration for stability.
6. *Final Monitoring and User Notification*: Continuous monitoring resumes once the new configuration is in place. The network status is assessed to ensure optimal performance, and a notification is sent to the user via the MCE, indicating successful recovery.
7. *LoT Recomputation*: The Trust Management system, using the LoTAF, continuously monitors network parameters related to the TLAs. After the reconfiguration, these parameters return to agreed thresholds, and LoT is recomputed and updated to reflect the restored trust levels.

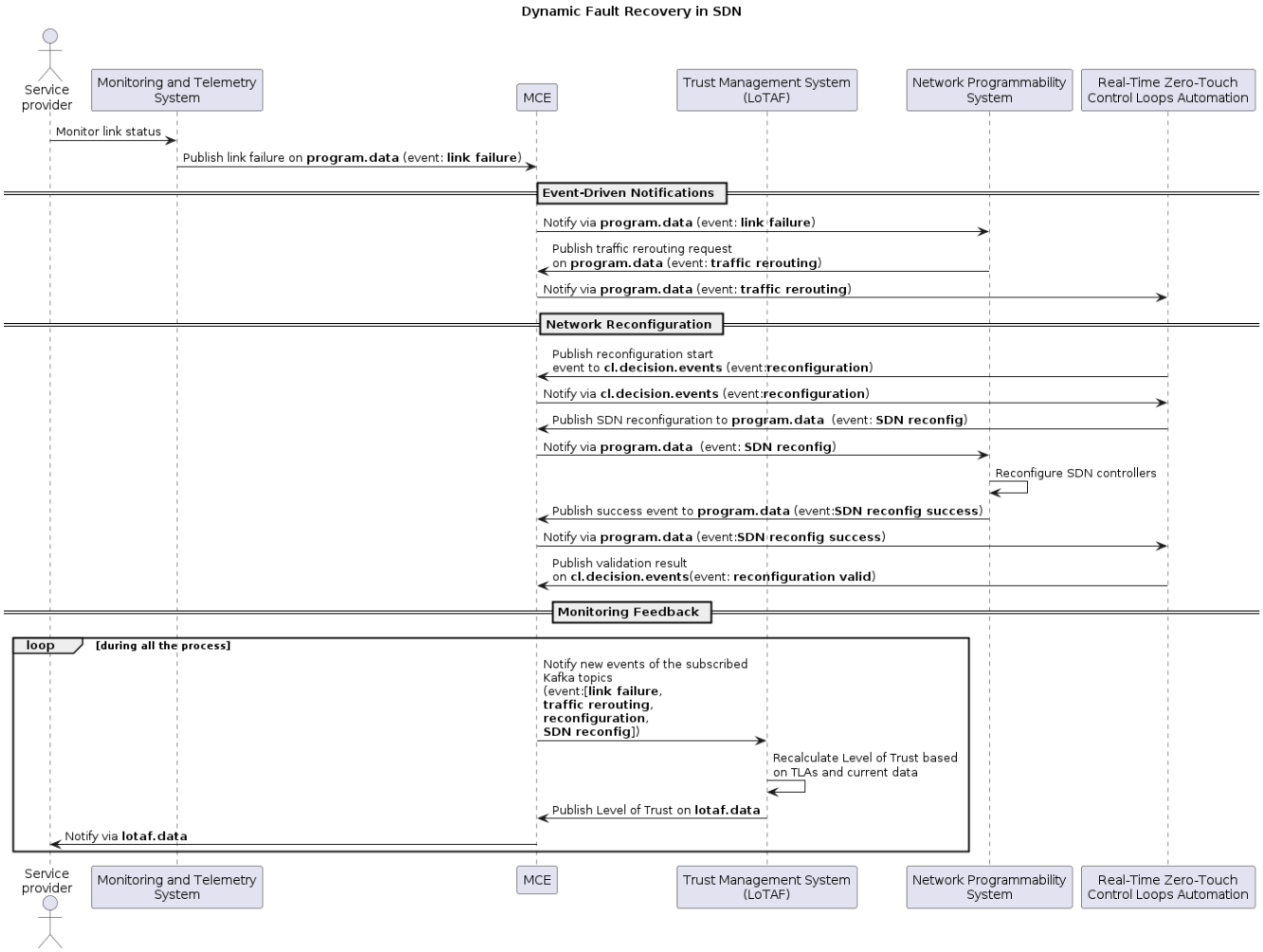

Figure 2-52: Dynamic fault recovery in SDN use case using MCE

Evaluation

The deployment of this implementation is currently in a state of active development while this document is written. The MCE itself has already been deployed on specific in-house testbed the integration with other components is in progress.

According to the Hexa-X-II work programme, to ensure a comprehensive and indicative assessment of the collective operations involved, this evaluation is planned to be conducted in the context of the project System PoC#C, that will be conducted under the scope of WP2 and reported in the deliverable D2.6.

In this context, the integration carried out through the MCE is planned to be subject to a thorough and systematic analysis, following the measure of KPIs declared in Annex A.4. The integration process is planned to be executed collectively, ensuring that all components of the smart management framework involved in this implementation are integrated. Adopting this holistic approach will enable to provide insights into the performance of the system as a whole. Furthermore, this approach is designed to showcase the enhancements and improvements introduced by the new integration methodology (i.e. an event driven approach for the M&O framework component coordination), demonstrating its impact on the overall system functionality.

2.2.2.2 Services orchestration over resources in the network continuum

Description

This section presents the integration of various enablers developed within the proposed smart network management framework to enhance energy efficiency, meet stringent latency requirements, and enable forecasting-based orchestration. The described implementation was realised within the scope of Component PoC#B.1 and supports deployments across multiple clusters and extreme edge infrastructure. It leverages Distributed Ledger Technology to facilitate federated service management in scenarios where centralised control of resources is not feasible.

Specifically, the implemented integration includes the following component enablers:

1. Certain components of the Multi-agent system for multi-cluster orchestration solution
2. Multi-agent Reinforcement Learning for adaptive scaling algorithm
3. Certain components of the Decentralised Orchestration solution
4. The SLA-driven Federated Orchestration functionality
5. The Efficient Network and Service Function allocation algorithm
6. The Sustainable MLOps system

Performance is evaluated through realistic experimental trials, utilizing an open-source, latency-sensitive distributed 6G application developed specifically for testing, combined with an infrastructure emulator that models the features of the envisaged future 6G networks. Experiments are conducted based on specific workflows of the described implementation and they are presented later in this section.

Implementation architecture

The implementation's design details are illustrated in Figure 2-53. The main components included are designed to encompass the functionalities necessary for making decisions, deploying services, and monitoring deployments within a single domain, while also enabling inter-domain communication. They are described below:

- *Application Descriptor*: This component provides an interface for the service provider to describe the services for deployment and their corresponding specifications, such as scalability, privacy concerns, and optimisation criteria. A descriptive language is necessary to express the services' computation and communication requirements. Therefore, to describe the application used for experimentation, the application graph model is used as proposed in [ZFV+23]. This description was selected because it effectively captures the requirements of highly distributed applications, including its connectivity specifications and constraints regarding specific services or workflows.
- *Infrastructure*: The infrastructure discussed in the context of a specific domain regards both network and computing resources. These include not only physical and virtual resources, but also emulated elements provided by the Infrastructure Layer Emulator (ILE) [ILE24] developed in the context of this Hexa-X-II project.

- *Deployment Manager*: The Deployment Manager supports single-cluster as well as multi-cluster service inter-connectivity. The technologies adopted in this component tackle the issue of connecting distributed clusters into a single domain, moving responsibility to a centralised point, allowing complex deployment schemes to occur by decoupling inter-connectivity issues from the decision-making process. Moreover, this component is responsible for applying orchestration actions to the infrastructure, such as scaling and migration actions, application deployment, and load balancing rules.
- *Orchestrator*: This component encompasses the orchestration intelligence responsible for any decision-making related to service deployment. The orchestrator's objectives include meeting the SLAs of registered services, optimizing key performance metrics, evaluating incoming service federation requests, and maintaining application workflows across domains. This multi-objective role demands sophisticated decision-making, for which AI-driven techniques are employed.
- *Monitoring Framework*: Information on service deployments is collected at this component by gathering data from both the services themselves as well as the resources they consume. Additionally, federated services require closer and more intricate monitoring to ensure compliance with their respective SLAs. Observability [TAZ+22] plays a crucial role in modern service deployment systems, where multiple data types (such as metrics, traces, and logs) are essential for effectively managing the deployment life cycle.
- *Federation Manager*: This is a DLT-based component enabling dynamic negotiation and execution of service federation. It facilitates efficient service deployment across multiple domains by leveraging smart contracts. This functionality is crucial for ensuring service continuity in dynamic scenarios. However, privacy concerns must be addressed when allowing services to operate in external domains, particularly for those registered with specific privacy requirements.

These components and the interactions between them are based on the *Multi-agent system for multi-cluster orchestration* (Section 2.1.1.1) and some components of the *Decentralised orchestration* (Section 2.1.1.2), and evaluate their applicability to real environments through actual implementation and experimentation with real applications and testbeds.

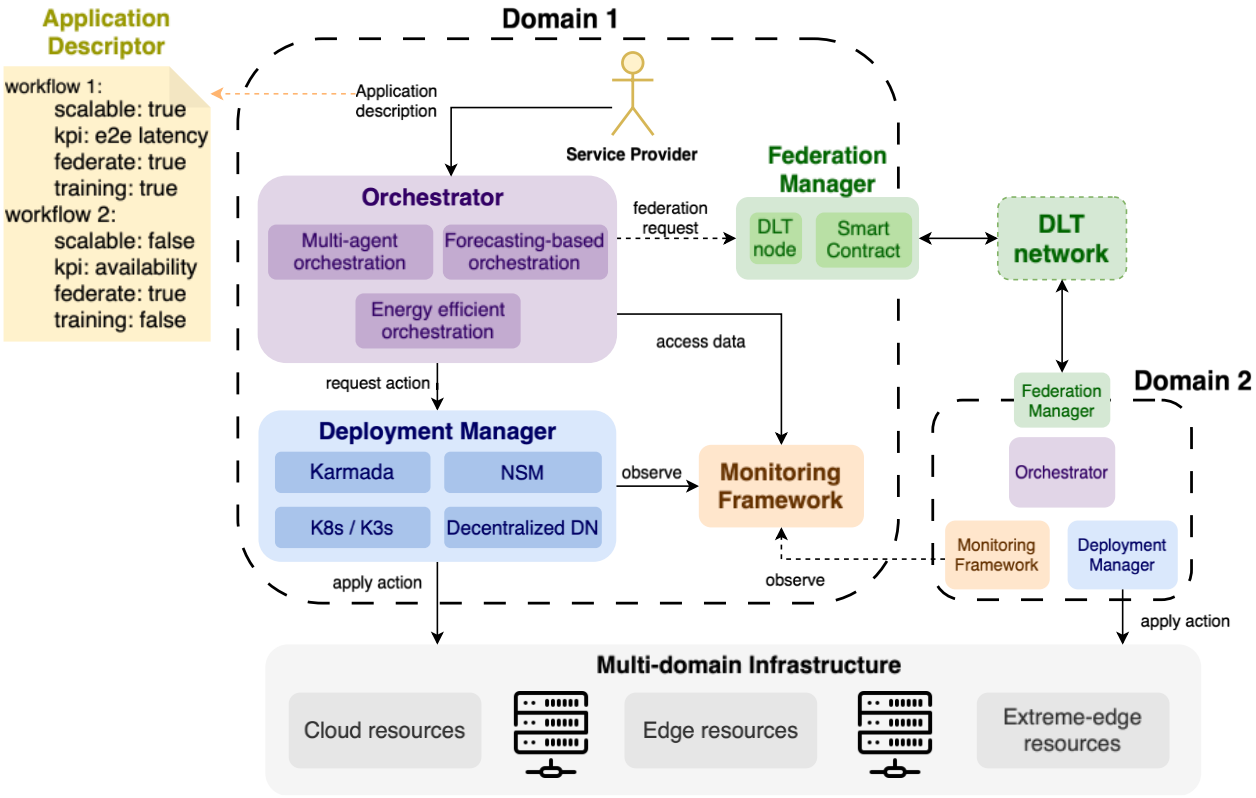

Figure 2-53: Implementation overview.

In Figure 2-54, the generic flow that the implementation components follow regarding their interactions is described, while specific workflows realizing specific functionalities on which evaluation is conducted are

presented after this. The implementation begins with the service provider registering the service constraints, the KPIs of interest, and inter-service communication requirements. Next, the orchestrator creates an initial deployment plan and instructs the multi-cluster manager to allocate microservices across various clusters optimally. Each service workflow, which may have unique requirements, is then mapped to appropriate orchestration mechanisms to ensure efficient management. A continuous service lifecycle management process is maintained through an online feedback loop, which iteratively determines orchestration actions related to horizontal scaling and migration. These decisions are informed by performance metrics stored and accessed through the Monitoring Framework. Finally, when the deployment spans multiple administrative domains, service federation is activated through the federation manager, ensuring seamless coordination across different domains.

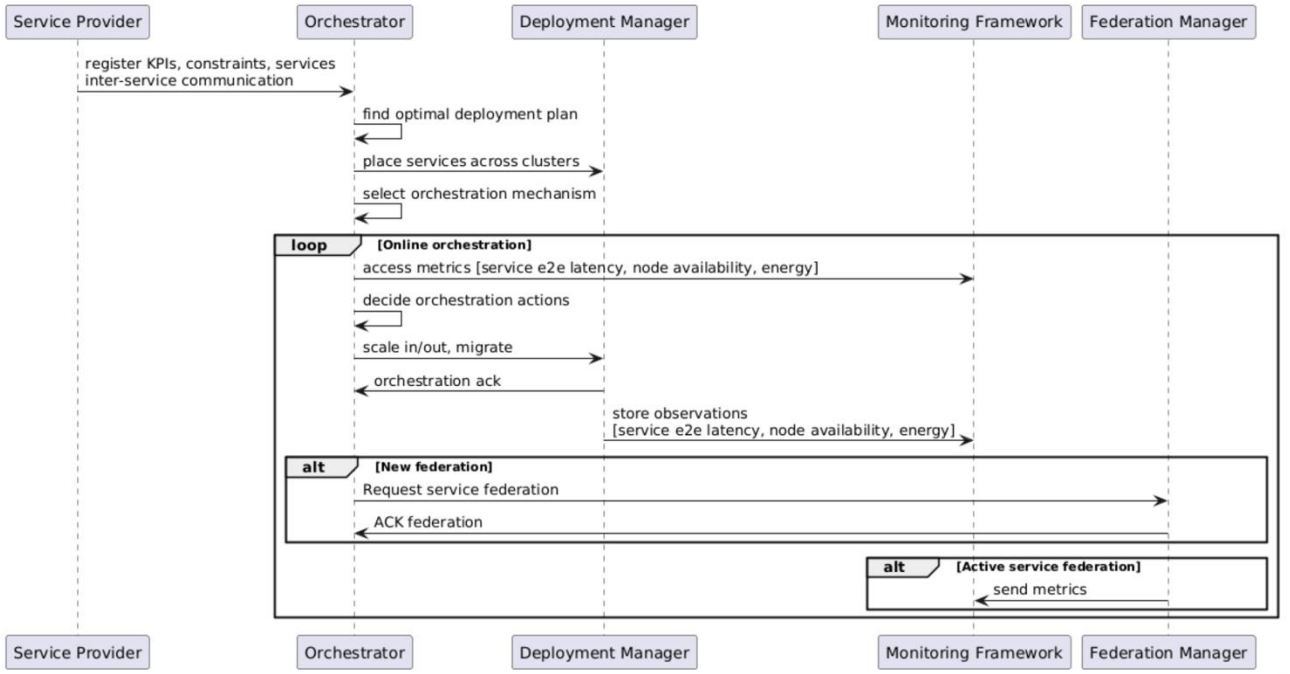

Figure 2-54: Generic orchestration workflow.

Evaluation service

In order to evaluate some of the M&O mechanisms of the management framework on the network continuum, an open-source microservice-based streaming application was developed within the context of Component-PoC#B.1. It revolves around smart manufacturing, explicitly focusing on real-time surveillance of manufacturing processes and ensuring safety through prompt responses to detected anomalies. The considered use case involves a stationary robotic arm performing specific functions whose operation must be automated. Additionally, a remotely controlled robotic vehicle allows a user to monitor the location. This remote driving feature includes streaming a video feed from a camera mounted on top of the vehicle to the operator, and transmitting remote control commands in the opposite direction. Another key aspect is the integration of machine learning-based object detection on frames sampled from the video feed. This functionality enables real-time data analysis and decision-making, triggering an alert to automatically pause the robotic arm when necessary.

Figure 2-55 displays a high-level view of the application's design and architecture. As illustrated, the provided functionality consists of two separate workflows regarding object detection and alert generation, which controls the robotic arm's operation. The main components of the remote operation workflow include a sender at the vehicle transmitting the native MJPEG video stream and an encoder at the edge that receives the video stream, encodes it to H264, and publishes it to the media server, which in turn distributes the newly encoded stream via WebRTC. Finally, the frontend at the remote operation centre serves as the interface with the human operator, displaying the received video feed and accepting the remote-control input that is relayed to the vehicle. The object detection workflow consists of a sampler at the edge that receives the same video feed from the camera mounted on top of the vehicle, samples frames at a dynamically adjustable rate, and provides them

as input to the object detector. The latter is built upon a pre-trained deep neural network suitable for real-time detection (i.e., YOLOv8 [YLO24]). Based on the results of this real-time object detection, an alert is generated if an anomaly is detected (e.g., when the presence of person is detected at the location), and the operation of the robotic arm is paused.

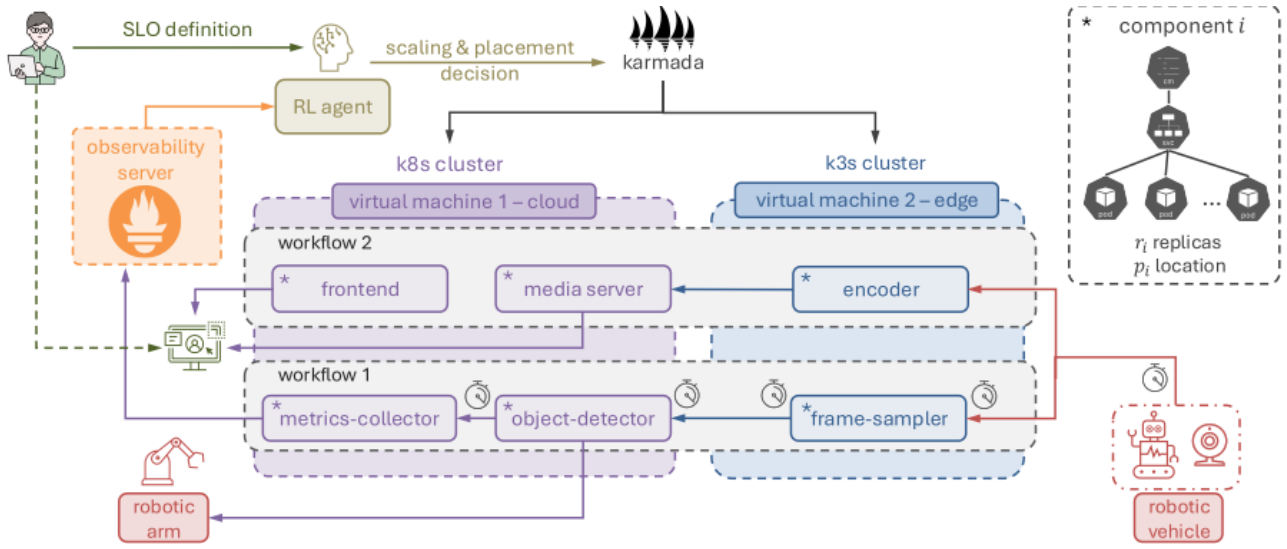

Figure 2-55: 6G latency sensitive service.

In the following sections, the specific workflows used for evaluation of the implementation are presented for a more fine-grained view of the implementation processes.

Multi-Agent Reinforcement Learning (MARL) scaling and migration

In the multi-cluster setup, a single operator may hold the responsibility to manage service deployment across multiple clusters, but the individual cluster operators may also deploy their own orchestration agents for consuming local resources. Such layered environments, depending on the agreement between operators can create various inter-agent dynamics. This workflow takes advantage of the proposed framework to setup different agent networks that collaboratively or independently try to optimally handle their given tasks based on their specific objectives. It exploits the Application Descriptor to identify the relevant KPIs, utilises the multi-agent orchestration mechanism of the Orchestrator and the service deployment capabilities of the Deployment Manager to deploy the service chain on top of the multi-cluster infrastructure. The Monitoring Framework is used for the real-time observation of the services' deployment during training of the agents and during evaluation of the management lifecycle.

Regarding the workflow's experimentation setup, the object detection service chain is orchestrated using the autoscaling and migration mechanisms described across the two clusters. Specifically, the ML-based object detector component is selected to be horizontally scaled as required based on the measured workload and end-to-end latency, being migrated from the edge to the cloud and vice-versa. During the service's normal operation, the sampling rate is low, meaning the traffic workload is low. In the examined scenario, the robotic arm is assumed to operate in a hazardous environment; therefore, the presence of people is considered an anomaly. While the detector tries to validate if there is a human in scope, the sampling rate is high, which introduces higher traffic in the service chain (warning operation). If a human continues to be detected, the robotic arm stops, and the sampling rate is reduced again to low (danger operation). When a human is no longer detected, the system returns to normal operation where the sampling rate and the traffic workload are low.

In the conducted experiments, a system-of-systems hierarchical approach is instantiated with a two-layer orchestration strategy that delegates responsibilities from top-layer systems to lower-layer ones, simplifying the management of distributed deployments. In this structure, the top layer corresponds to the multi-cluster manager, responsible for deciding service placement across clusters. On the other hand, the lower layer consists of per-cluster orchestrators that handle microservices' scaling by identifying the appropriate scaling factors towards optimizing for the designated objectives. The hierarchical mechanism uses a MARL methodology where each agent tackles a different task: agent 1 migrates the service across clusters and agent 2 handles the scaling of the service. This mechanism is compared with a centralised, RL-based, joint autoscaling-placement

mechanism managing the application. The orchestration actions of both setups are decided based on the evaluation of real-time data regarding delays (reflecting computation and communication latency) collected at specific monitoring points. The autoscaling and placement mechanisms evaluate the collected data, and for every interval, the agents select where the optimal placement is (taking into consideration resource and migration constraints) and how many replicas are needed for guaranteeing Service Level Objectives (SLOs) satisfaction in terms of end-to-end latency requirements.

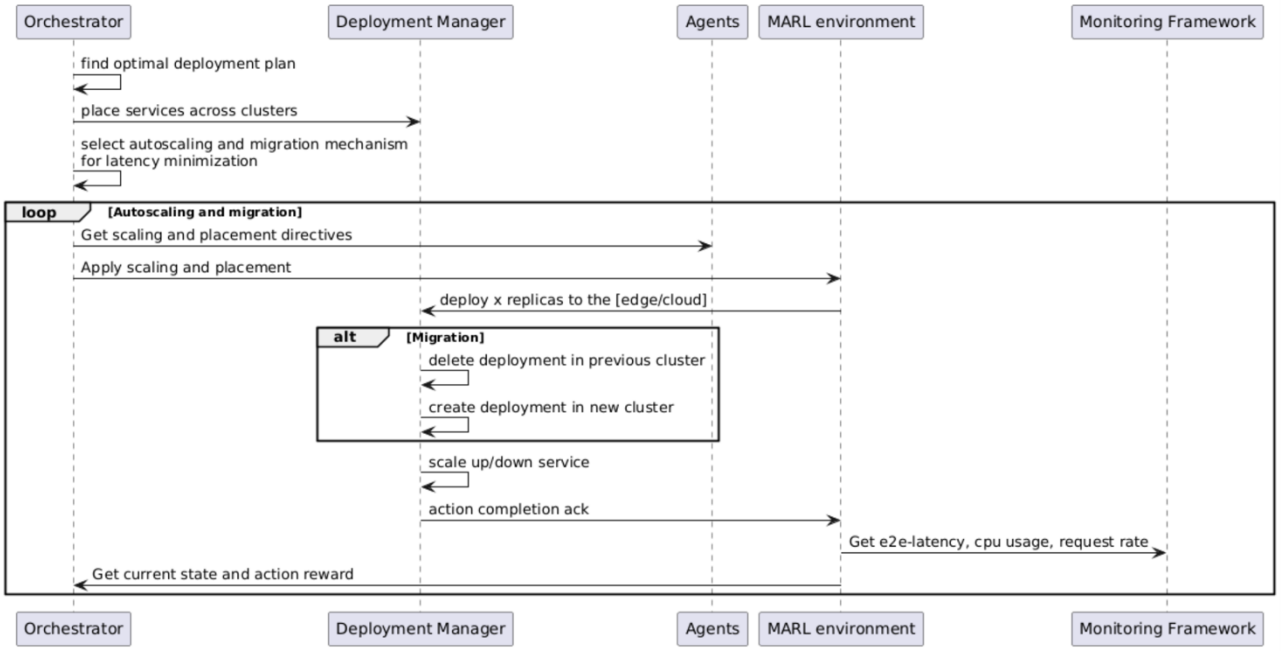

Figure 2-56: MARL scaling and migration workflow.

Three different agent setups are used to demonstrate how different stakeholder groups can influence SLA satisfaction: i) Joint Scaling - Migration (JSM), ii) Independent Scaling - Migration (ISM), iii) Mixed Scaling - Migration (MSM). JSM assumes a central Deep Q-Network (DQN) agent taking joint scaling and migration decisions, ISM assumes independent scaling and migration by two DQN agents and MSM tries a hybrid approach where agents act independently but consider global rewards as feedback adopting the QMIX [RSS+18] methodology. The results are illustrated in Figure 2-57. Specifically, the percentage of requests that violate the SLA (i.e., the latency threshold), the utilisation of the link between the two clusters, and the resource consumption in terms of percentage of used replicas are depicted for the three different setups.

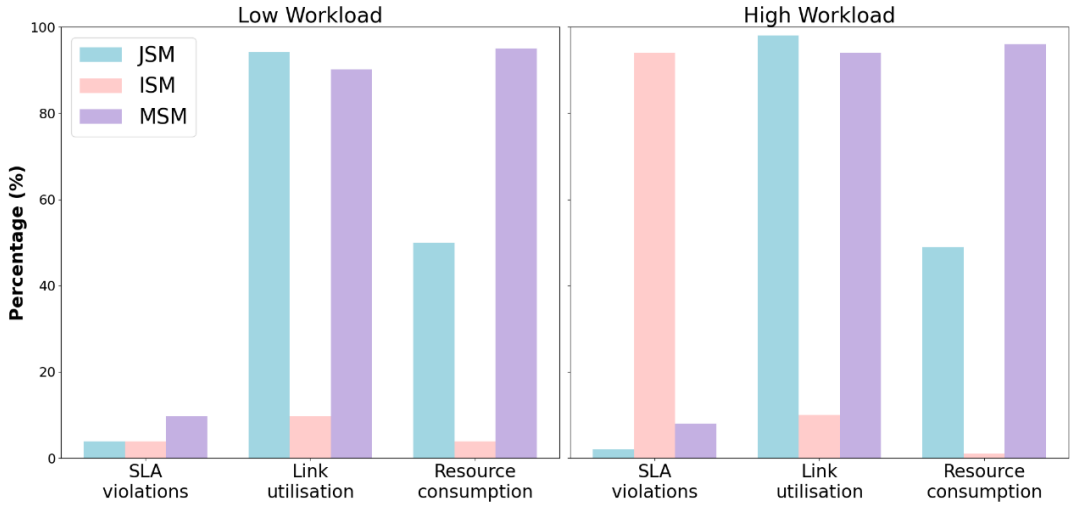

Figure 2-57: SLA fulfilment.

JSM meets the SLA requirements for both low (1 frame/sec) and high (10 frames/sec) workloads. However, this comes at the cost of high link utilisation and medium to high resource consumption. On the other hand,

ISM demonstrates minimal link utilisation and resource consumption but fails to satisfy SLA requirements for high workloads, as its agents operate independently. MSM, in contrast to ISM, introduces collaboration between agents, reducing SLA violations but increasing link utilisation and resource consumption, even for low workloads (1 frame/sec). The comparison between ISM and MSM highlights how introducing collaboration directives can enable agents operating in a decentralised manner to achieve global objectives, albeit with higher resource costs. Furthermore, while JSM outperforms MSM in this relatively simple setup with only two orchestration tasks, MSM's advantage lies in its ability to scale more effectively to complex scenarios. Finally, centralised solutions like JSM may not always be feasible when multiple stakeholders are involved.

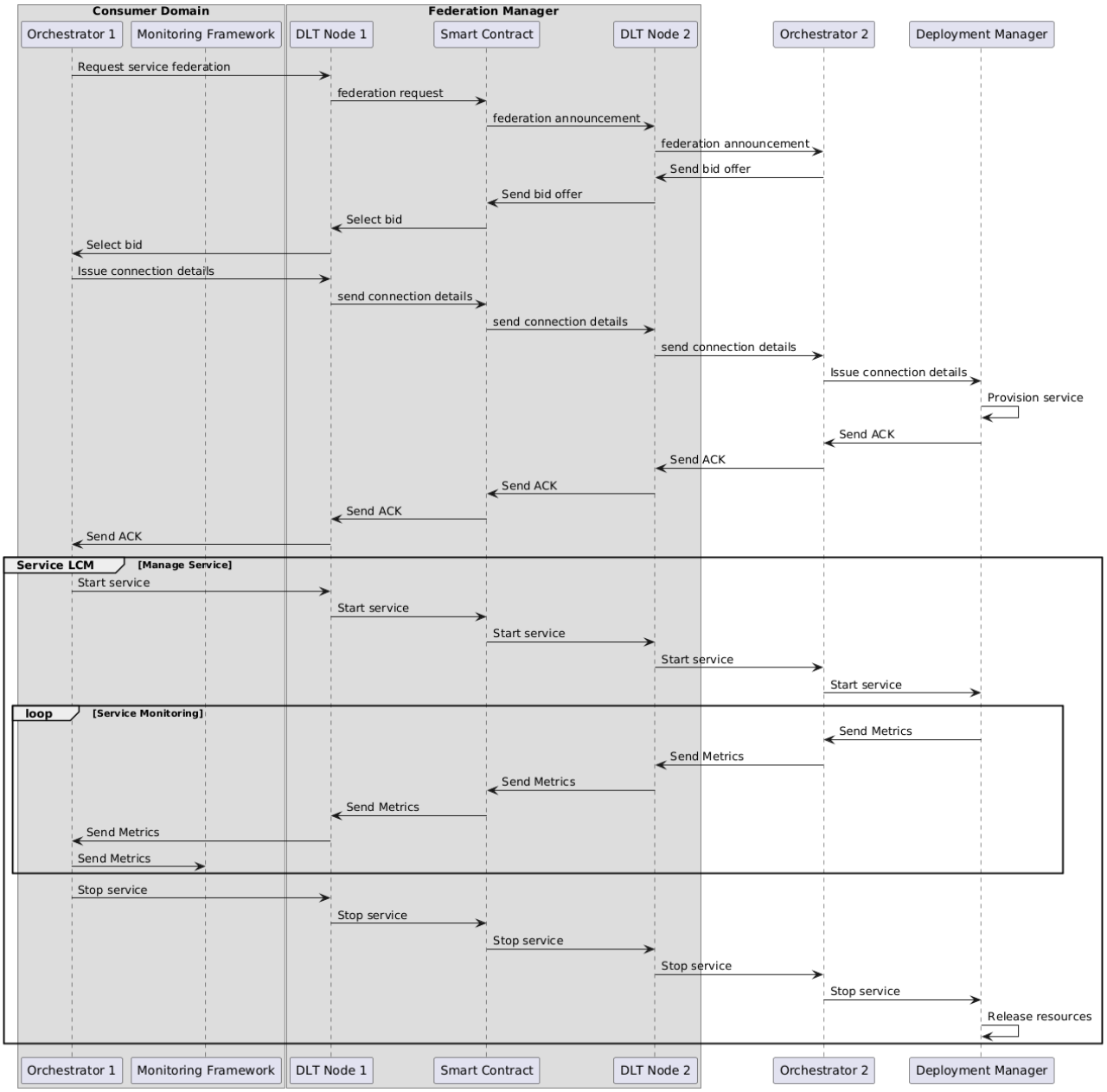

Figure 2-58: Service federation workflow.

Service federation

The workflow for the DLT based federation is depicted in Figure 2-58. The process is triggered by the orchestrator in the consumer domain in response to the needs of a running service experiencing degraded QoS. The DLT broker node receives the service KPI requirement and triggers the smart contract to send out federation request to participant domains. The domain with the best bid offer is selected and the connection

details sent to it in order to re-establish the service in this new provider domain (Domain 2). Once the service is provisioned in the latter, an acknowledgement is sent back to the consumer domain so that the service can be started. During the lifecycle of the service, the monitoring framework in the consumer domain receives updates of service metrics. The consumer domain maintains the role of life-cycle management throughout as such is responsible for issuing the directive for service termination. On receipt of the latter directive, the deployment manager in the provider domain stops the service and releases the resources it assigned to it.

Sustainable MLOps

This workflow (see Figure 2-59**)** shows how the functionality of the Sustainable MLOps (S-MLOps) system has been integrated in the context of this implementation. As it can be appreciated, the S-MLOps asset is used to deploy on the simulated operator scope the AI/ML Model on which the so-called Infrastructure Status Prediction Module (ISPM) relies. As an MLOps workflow, this is done in a continuous way in an infinite loop, using the CI/CD DevOps approach.

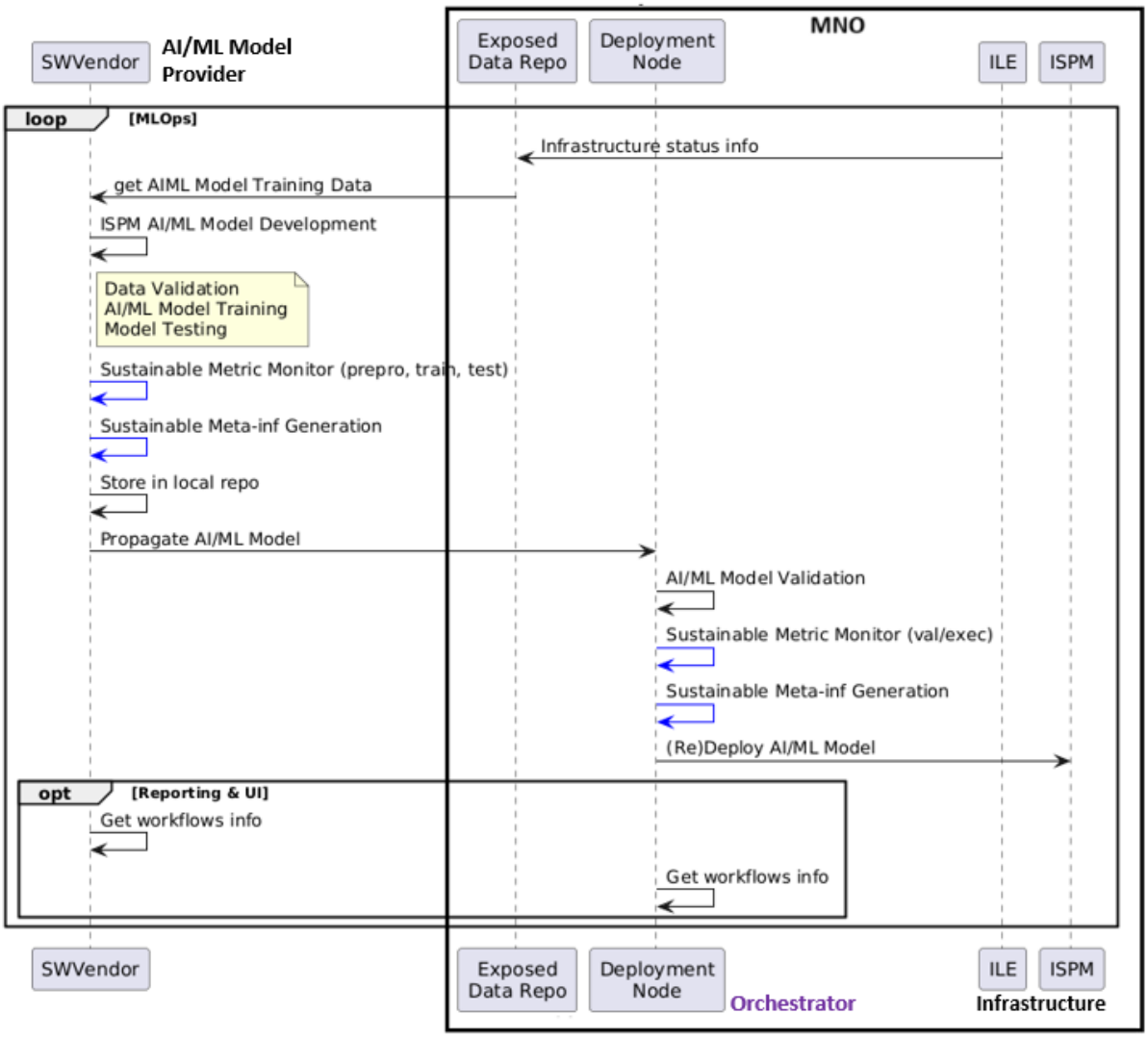

Figure 2-59: S-MLOps Workflow.

As it can be seen, the S-MLOps asset is used to develop and train the AI/ML Model in an emulated SW Vendor scope, using training data coming from the Infrastructure Layer Emulator in the MNO scope through an Exposed Data Repository, also in the MNO scope. The training dataset comprises historical time series collected from previous emulation runs conducted using the ILE. Specifically, the data consist of a Boolean value for each emulated node, indicating whether its status is 'on' (True) or 'off' (False). The sampling interval for each time series entry is 30 seconds. The Exposed Data Repository has been implemented using a TimescaleDB [TDB24] database, which exposes the data through the so-called Dataset Sharing Open API.

Also, as it can be appreciated, certain energy consumption related measurements are performed during the AI/ML Model training process in the SW Vendor scope. They are those processes labelled in blue in Figure 2-59. Specifically, those measurements are performed using:

- Scaphandre [SCP], a metrology agent dedicated to electric power and energy consumption metrics.
- Kepler, a Cloud Native Computing Foundation exporter for energy consumption estimations on Cloud environments.
- A CO_2 equivalent tracker, implemented as a custom agent for delivering information based on compute resources location and the corresponding energy production source distribution.
- A custom service designed to map the resources utilised during the execution of Kubeflow Pipelines tasks, providing detailed information aligned with the MLOps stages.

As it can be observed, each time the AI/ML Model is trained it is propagated from the SW Vendor domain towards the MNO scope, and specifically through the so-called Deployment Node (DN), on which the model is validated and served. Specifically, this DN has been implemented using the Model Sharing Open API to fetch the AI/ML model as well as the [TS] module to serve it in production. As illustrated in Figure 2-59, this validation process goes also hand in hand with energy-related measurements (blue arrows), in this case consisting of the monitoring of the energy consumption and CO_2 equivalent generation of all the processes involved in the MLOps workflow, providing the required sustainable metainformation linked to the different stages to be added to the ML resources shared between providers and MNO. Then, once the AI/ML Model is validated, it is deployed as part of the ISPM component.

As mentioned, this process can be repeated once and again in a continuous way, each time it is considered a new version of the AI/ML Model should be generated. Also, both SW Vendor and MNO can access the reporting and the provided GUIs to monitor the energy measurements in the different stages. The screenshot in Figure 2-60 shows the measurements associated to one of the deployment stages, where the different stages of an MLOps workflow with the respective consumption of the pipeline execution can be observed, taking into account all the resources involved (specific and common ones, like DBs or managers).

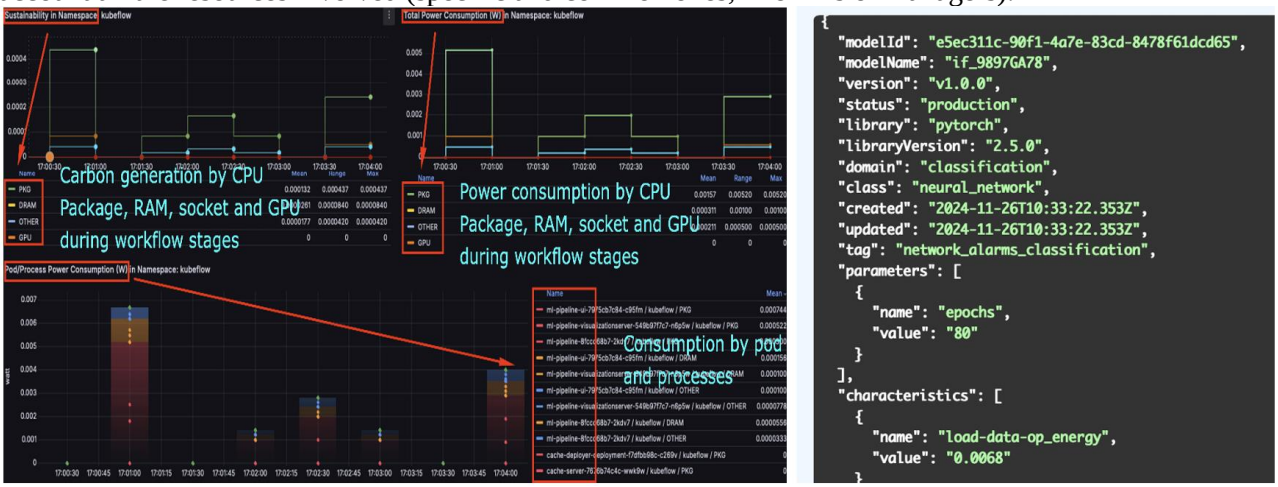

Figure 2-60: Workflow and energy consumption measurement illustration with sustainable metainformation available in model sharing API.

Proactive forecasting

This workflow (Figure 2-61) illustrates the proactive M&O functionality in this implementation relying on the ISPM and the Service Orchestrator (which functionality is implemented by the Availability Forecasting component in Figure 2-61). This ISPM is one of the components proposed from the Decentralised M&O system, in charge of predicting infrastructure status changes based on network data analytics. As it can be observed the ISPM is continuously fed with infrastructure status metrics taken from the ILE and generates infrastructure status predictions relying on its inner AI/ML Model (the one deployed through the S-MLOps workflow described before). Those predictions are communicated to the Service Orchestrator which, if necessary, generates re-location commands for those network service components that could be impacted by the forecasted disconnection of the nodes on which they were deployed.

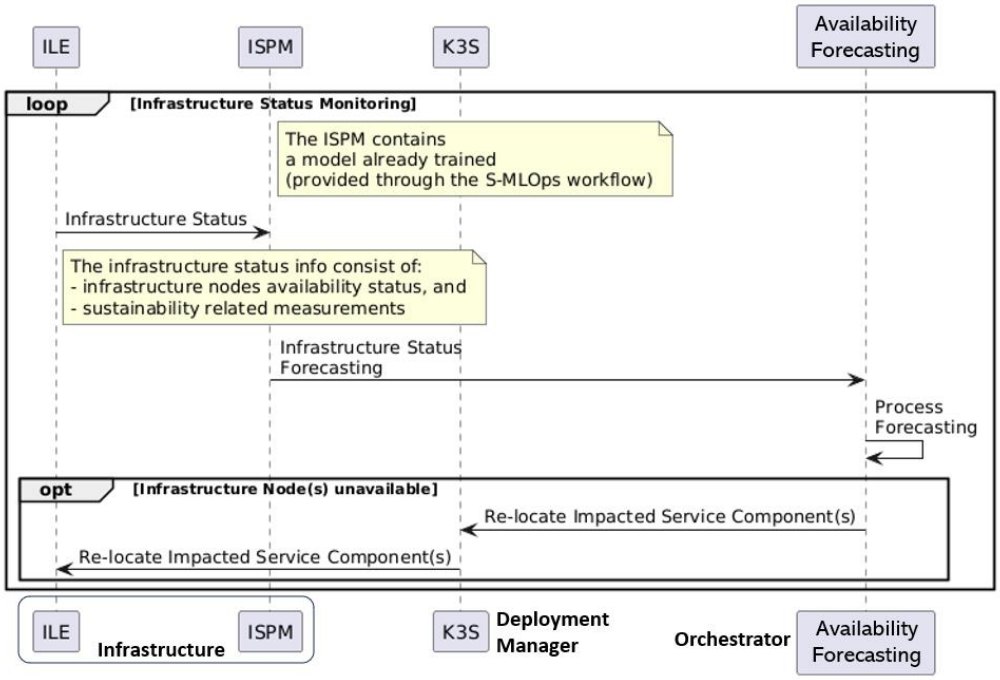

Figure 2-61: Proactive forecasting.

For this specific implementation the infrastructure metrics taken from the ILE convey information about the availability status of the infrastructure nodes (connected/disconnected), and also, about the battery status of certain devices at the extreme-edge domain which have been configured to behave as battery powered (i.e., simulating battery charge and discharge cycles). This aims to replicate the high-volatility characteristics of this extreme-edge domain, showcasing the proactive M&O functionality designed to maintain service continuity, even under highly volatile conditions, and where the service deployment status must dynamically adapt to changes in the infrastructure.

Specifically, the ILE is built on [LXD], a container and virtual machine manager that allows for running complete Linux systems within containers or virtual machines. It supports a broad spectrum of Linux distributions, ranging from lightweight versions suitable for deploying numerous nodes on limited hardware, to more robust distributions capable of handling complex network services. LXD is scalable, supporting configurations from a single instance to a cluster that simulates a full data centre, making it highly versatile for various use cases. For this implementation a setup consisting of 35 nodes was configured, mixing Ubuntu and Alpine Linux distributions.

Figure 2-62 illustrates an execution example where the ILE (the window with the green/red dots representing the available/unavailable Linux machines) hosts three components of the latency sensitive media streaming service described above (those in charge of processing the video streams), which as it can be appreciated, are deployed on three volatile extreme-edge nodes (blue line). As shown, when the ISPM forecasts the disconnection of the nodes on which the service is deployed, the service components are proactively migrated to other available nodes. During the experiments, the video streaming service used for the tests was repeatedly migrated without perceptible interruptions in the playback.

For the experiments, a pre-trained single feed-forward neural network with two distinct input pathways converging in the last hidden layer of the network was used. This network topology was chosen to enhance modularity and effectively handle the different nature of the data input: the continuous time values and the discrete network infrastructure state data information. The model was trained using a dataset containing historical data on infrastructure node availability over a one-week emulation period. This dataset was divided into two subsets, with (a) the infrastructure state at a specific time, and (b) its state three hours later (b). This enabled the model to predict the infrastructure status within this time frame. Once trained, the model triggered the proactive migration of the service components running on nodes predicted to go offline, ensuring high availability and service continuity. The ILE was configured to follow pseudo-random regular patterns, mimicking real networks and aiding the AI/ML model in generating reliable predictions. The migration

mechanism was implemented relying on the K8S taint command. Specifically, the experiment was set up to generate predictions every 30 seconds, while the network service migration time was between 4 and 6 seconds.

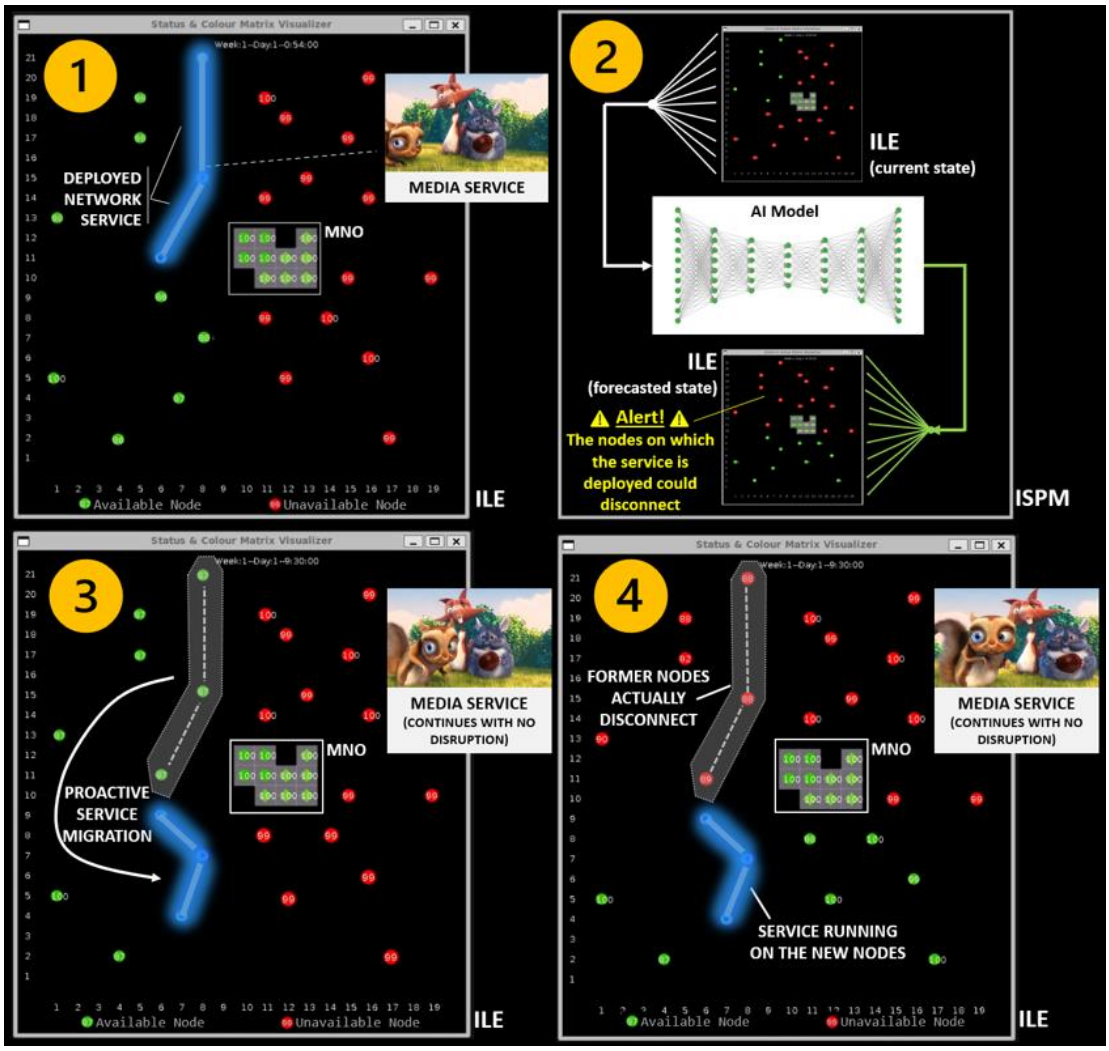

Figure 2-62: Proactive forecasting workflow execution example.

Alternatively, another migration mechanism was developed to select the target node for migration in a more intelligent and energy aware manner, where an AI/ML-based model was trained to select the migration target using energy related information as input such as the battery level of the extreme edge nodes that is provided by the ILE. When the need for migration is determined (i.e. when a node is predicted to go offline by the ISPM), the model takes as input the node status information (i.e., availability, resource capacity, battery level) which has been retrieved from the ILE, and provides as output the ID of the node that the service should be migrated to. The migration is then triggered by setting the node affinity of the service container in K8S to the label corresponding to the selected target node via an API call.

The model was trained by using periodical status information from the extreme edge devices in the ILE, and the decisions provided by the model were evaluated based on their impact on service performance and energy metrics (e.g., placing a service on a node with little battery left will result in another migration shortly after, and should be avoided). Additionally, to minimise the end-to-end latency for the network flow, the extreme edge nodes that are closer to the source node of the service are preferred.

2.2.2.3 Functionality allocation in a cobot-powered warehouse inventory management

The implementation described in this subsection focuses on showcasing management mechanisms including smart network management advancement. Specifically, a resource domain consisting of drones/unmanned aerial vehicles (UAVs), autonomous mobile robots (AMRs) and servers is used as a testbed for developing

and testing a beyond state-of-the-art automated inventory management audit solution for accurate and efficient warehousing operations.

Some of the essential technologies and components of this implementation include the advanced fusion of computer vision and sensor data for verifying the correct identification, counting, and real-time location of objects. Another key component is the dynamic translation system between symbolic warehouse locations and 3D geometric coordinates, enabling smooth drone and AMR navigation along with precise inventory pinpointing. Additionally, the M&O component, referred to as "functionality allocation", is also integrated in this scenario, and is responsible for the optimal placement and planning of computational and physical workloads and robotic tasks.

One of the main scenarios studied is the optimal placement of the various inventory management services (e.g., item scanning cobot-role) and workloads requiring considerable computational resources (e.g., computer vision tasks). The placement considers factors like current workload, energy availability (for mobile, battery-operating devices), hardware capabilities (e.g., ground/aerial node), and physical environment parameters, such as real-time proximity to the inventory locations.

Key services/workloads/roles for this scenario include object detection, path planning, inventory management, quality inspection, and the warehouse digital twin (a 3D real-time model representing the warehouse environment). The system architecture, depicted in Figure 2-63, encompasses components for orchestration and monitoring, AI domain resources (such as the energy efficient functionality allocation algorithm), inventory management-specific services (such as object detection and path planning), user interfaces, and network domain resources.

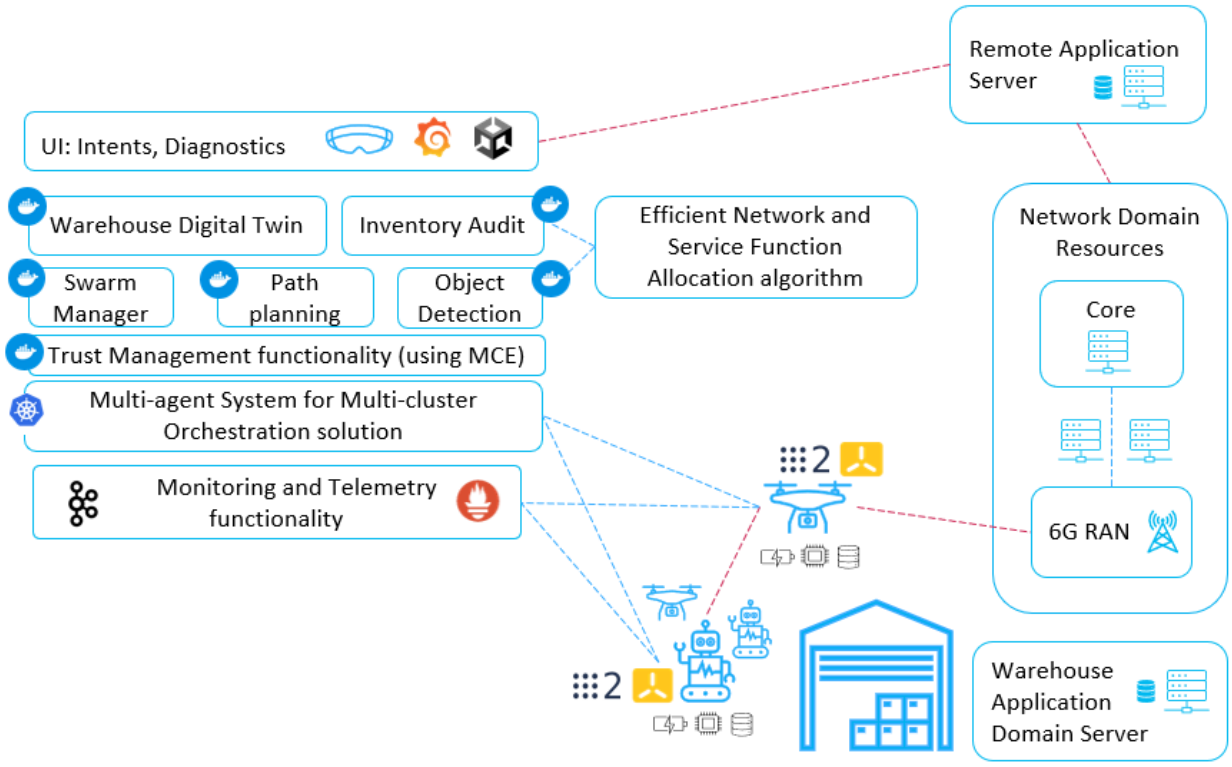

Figure 2-63: Schematical representation of the platforms and components of the cobot-powered warehouse inventory management implementation.

The main M&O enablers included in this implementation are:

i. The Efficient Network and Service Function Allocation algorithms, with the functionality allocation joint optimisation mechanism which considers computing and transmission energy for computational workload placement and physical task planning.
ii. The Trust Management functionality, which includes the TEF and LoTAF functions, and evaluates the trustworthiness index per entity of interest in the specific scenario, being considered when functionality placement occurs.

iii. The Multi-agent System for Multi-cluster Orchestration solution, where the orchestration process of the functionality allocation mechanism is analysed for ensuring the successful deployment of network services and applications, as well as the orchestration enforcer.
iv. The Monitoring and Telemetry functionality, for collecting infrastructure KPIs and metrics.
v. The MCE functionality for multi-stakeholder support mainly used between TEF and LoTAF functions within trust management functionality (see Section 2.1.2.5).

With AI-assisted, trust- and energy-driven optimisation, this configuration aims to demonstrate robust and reliable operation scenarios in warehouse and manufacturing environments, considering compute continuum node performance, energy availability, and network reliability.

This implementation has been integrated in the Hexa-X-II PoC#A/B/C. As it is ongoing work, updates and additional results are planned to be reported in D2.5 and D2.6.

Workflows

The overall messages sequence diagram of the implementation is shown in Figure 2-64. As it can be appreciated, beyond the continuous collection of KPIs and status metrics, the operational trigger starts with an intent towards the so-called API Server (within the Multi-agent System for Multi-cluster Orchestration solution) which is an interface component of the orchestration manager responsible for checking the feasibility of the various requests and responses coming from most of the components (monitoring and telemetry framework, trust management system, and energy efficient resource allocation component). It receives intent-based requests and triggers the appropriate components (e.g., monitoring and telemetry framework, trust management system, energy efficient resource allocation) for accomplishing a task. The infrastructure monitoring collects various metrics, and information related to the status, GPU/CPU, memory, storage availability of the physical and virtual resources of the system, checking also that some thresholds are respected, and the Service Registry stores the information and requirements of the various computational workloads and tasks, both related to the Monitoring and Telemetry functionality. Finally, the orchestrator, as the orchestration enforcer indicates, enforces the placement decision to the system's resources (within the Multi-agent System for Multi-cluster Orchestration solution as well).

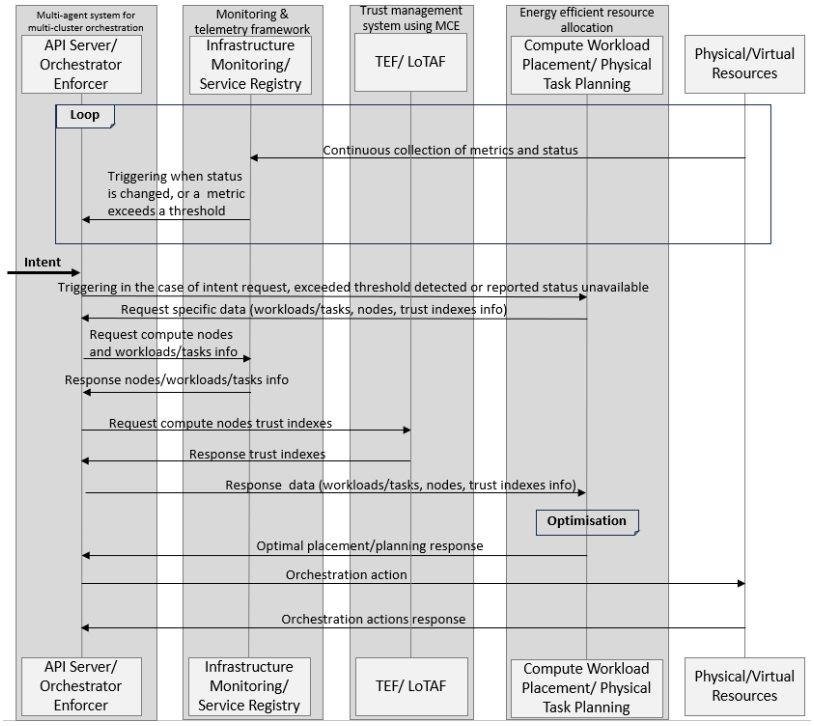

Figure 2-64: Workflow of the cobot-powered warehouse inventory management implementation.

As shown in Figure 2-64, the monitoring and telemetry framework continuously collects and updates the various KPIs and status metrics of the system's nodes, so that if a metric exceeds a threshold or the status of a device becomes unavailable, it triggers so towards the API Server. Based on this, when the API Server receives

a request from the infrastructure monitoring, or an intent-based request coming from the user, it triggers the Energy Efficient Service Function Allocation-functionality allocation. The functionality allocation mechanism requests from the API Server specific information related to the compute nodes capabilities, the computational workloads, and tasks requirements as well as the compute nodes' trust indexes. The API Server accordingly asks this information from the monitoring and telemetry framework and the trust management functionality and, as soon as it receives the responses, it provides them (as input) to the Efficient Network and Service Function Allocation - functionality allocation algorithm. Then, the functionality allocation mechanism performs the optimisation process utilising both the compute workload placement and the physical task planning algorithm. When a proposed placement solution is produced, it is sent to the API Server. The API Server checks then the solution and asks the decided action from the Orchestration Enforcer, which in fact enforces the placement/planning decision and receives a response as soon as it is completed.

Evaluation

Some initial evaluation results regarding this implementation have been already reported in preceding deliverables, specifically, in [HEX223-D22] some simulated results of the early version of the computational workload placement functionality allocation algorithm was presented showing 8.8-28.6% power consumption gains (related to OPEX and energy efficiency KPIs) when comparing the developed metaheuristic mechanism (algorithm based on the genetic algorithm paradigm [KCK21]) to the round-robin placement. Also, in [HEX224-D23] the performance of the improved version of this algorithm was reported by measuring the execution time of the algorithm and the score (value obtained by the minimisation of the objective function which consists of an energy consumption term, an E2E latency term and a trustworthiness related term) for increasing number of computational workloads. These measurements showed that the developed functionality allocation mechanism had close to optimum scores within significantly less time than the PuLP GLPK solver [MOD11] as the number of workloads increased (related to scalability KPI). Moreover, in [HEX224-D63] trustworthiness and energy consumption measurements were collected using simulated data for increasing numbers of computational workloads and varying trust and energy weights (weights of the multi-objective function presented in Section 2.1.4.2). Specifically, the trustworthiness was defined as the sum of the trust indexes of the compute nodes utilised for the workload placement. Additionally, the energy consumption was defined as the sum of the processing and. transmission energy consumption of the compute nodes. The simulation results showed up to 43% increase of trustworthiness (computed as the sum of the trust indexes of the compute nodes utilised for the placement, which were expressed as the weighted sum of relevant KPI values, such as availability, reliability, and security, among others) and up to 50.9% energy consumption gains (related to OPEX and energy efficiency KPIs) of the metaheuristic developed mechanism compared to round-robin placement.

Beyond these initial results, new results were further obtained, specifically related to the energy consumption and the total duration time of completing all tasks when using the physical task planning functionality allocation algorithm (see Section 2.1.4.2). For this, thirty fixed virtualised AMRs and UAVs were utilised, with various capabilities and an increasing number of physical tasks (computer vision tasks) in varied warehouse locations. In particular, the AMRs used moved at 0.7-1 m/s mean speed, with 60-80 W mean travel power consumption, and 30-40 W mean power consumption when executing the stationary task (computer vision), while the UAVs moved at 7-8 m/s mean speed, with 150-230 W mean travel power consumption, and 100-120 W mean power consumption when executing the stationary task (also computer vision). The time to execute each task was assumed for both AMR and UAV to be 3s (which is the mean value of actual measurements). The tasks were distributed in a warehouse with dimensions of 20m length, 20m width, and 4m high. In these measurements/experiments the initial number of ant colony optimisation - ACO's ants was 200, the number of best ants was 40, the decay was 0.99 (this is the rate that pheromone matrix is multiplied with so that old pheromones does not confuse next generations ants), alpha was 1.5 (alpha acts as weight on the pheromone), the beta was 3 (beta acts as weight on the objective function) and it was used a stopping criterion of maximum 20 iterations having no improvement.

Figure 2-65 shows the total energy consumption and the energy consumption gains obtained by using the ACO-based task physical planning functionality allocation algorithm compared to the nearest neighbour heuristic (baseline) for increasing number of tasks (from 10 to 120). As it is shown, the proposed algorithm obtained up to 35.9% energy consumption gains (related to OPEX and energy efficiency KPIs). As the number of tasks increase, the gains in energy consumption increase, until a critical point, at which the ratio

robots/number of tasks becomes smaller than 0.5. In other words, the described solution provides higher gains when there is sufficient availability of robots, thus solution space and optimisation potential. The scarcer those resources/robots become, the lower gain potential is observed.

Finally, Figure 2-66 shows the total duration time of completing all the tasks and reduction of duration time by using the ACO-based task physical planning functionality allocation algorithm compared to the nearest neighbour heuristic (baseline) again for increasing number of tasks (from 10 to 120). As shown, the proposed algorithm achieved up to 60% reduction in duration time (related to scalability KPI). However, the gains gradually decreased as the number of tasks increased, due to the scarcity of robots/resources which resulted in lower gain potential.

This work is still in progress; hence the final results will be provided in the context of WP2 in D2.5 and D2.6, according to the project work plan.

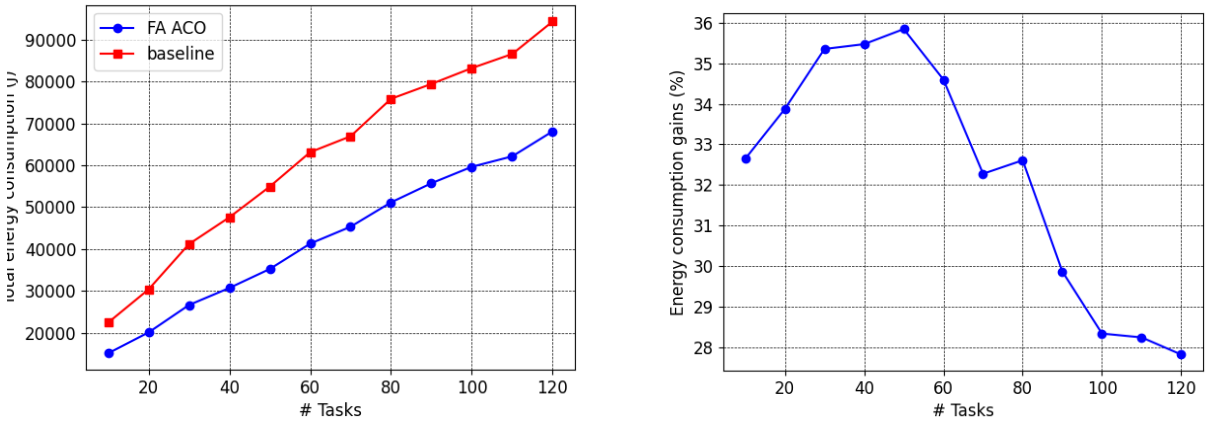

Figure 2-65: Total (left) energy consumption and the gains (right) using the physical task planning algorithm based on ACO algorithm compared to the nearest neighbour heuristic (baseline).

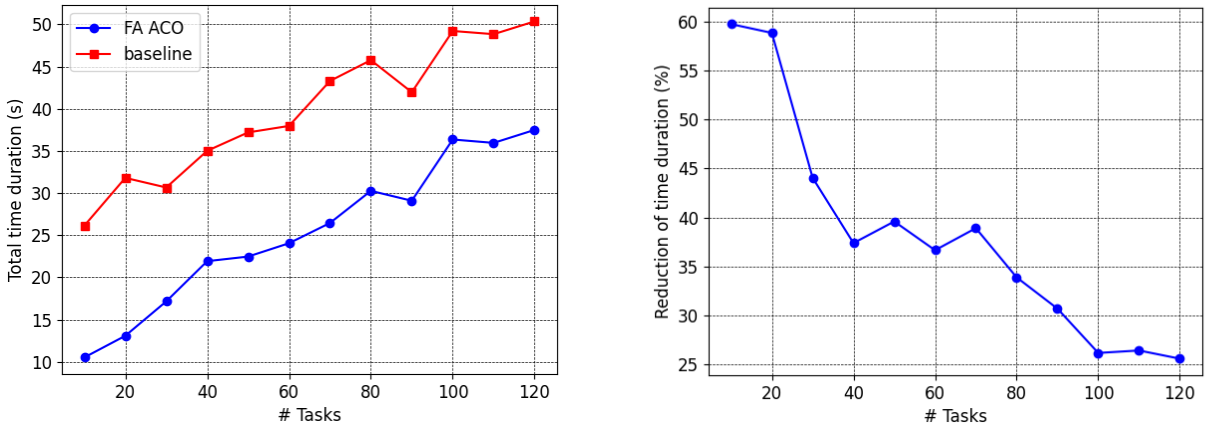

Figure 2-66: Total (left) duration time and the reduction (right) using the physical task planning algorithm based on ACO algorithm compared to the nearest neighbour heuristic (baseline).

2.2.2.4 ML-based recommendation for energy management in service orchestration

Description

It is well-known that the radio access network (RAN) accounts for approximately 80 to 85 percent of overall energy consumption of mobile networks. Considering geographical variations and fluctuating data traffic loads, it is advisable to put some capacity cells into sleep mode and activate them based on traffic demand. This approach acknowledges that not all cells need the same amount of energy to meet traffic requirements due to differences in user and network activity. This scenario calls for a customised strategy for each capacity

cell to determine whether it should be in a sleep or active state. Implementing machine learning (ML) and artificial intelligence (AI) techniques enhances the potential for such energy-saving opportunities across the network at the cell level. This implementation aims to have a dynamic threshold configuration for cell sleep modes to effectively manage both coverage and capacity.

To enable and control cell sleep mode based on Physical Resource Block (PRB) utilisation and Radio Resource Control (RRC) connection thresholds—it is considered crucial to carefully monitor the performance indicators such as network availability, reliability, traffic patterns, services offered, and spectrum usage, including those of neighbouring cells. Achieving this requires determining the utilisation of each cell layer for the upcoming days and, most importantly, assessing the potential impact on customer experience during that period. In the current scenario, static thresholds that are manually set through the cell sleep mode feature are considered. However, one can change these thresholds dynamically. To determine the values for these dynamic thresholds, it was necessary to conduct field experiments on a live network, as there was no variation in data due to the static nature of the existing cell sleep mode. The primary target of this implementation is to help have zero-touch management framework that autonomously monitors network performance and dynamically adjusts network configuration (in this case cell sleep mode) without needing human intervention to save energy in the network. The implementation can interact with the two main components. First, it can interact with Monitoring and Telemetry functionality to monitor the required data. Second, it also can get help from RL zero-touch CLs functionality to automise the process and avoid any possible conflict. The workflow details are shown in Figure 2-67.

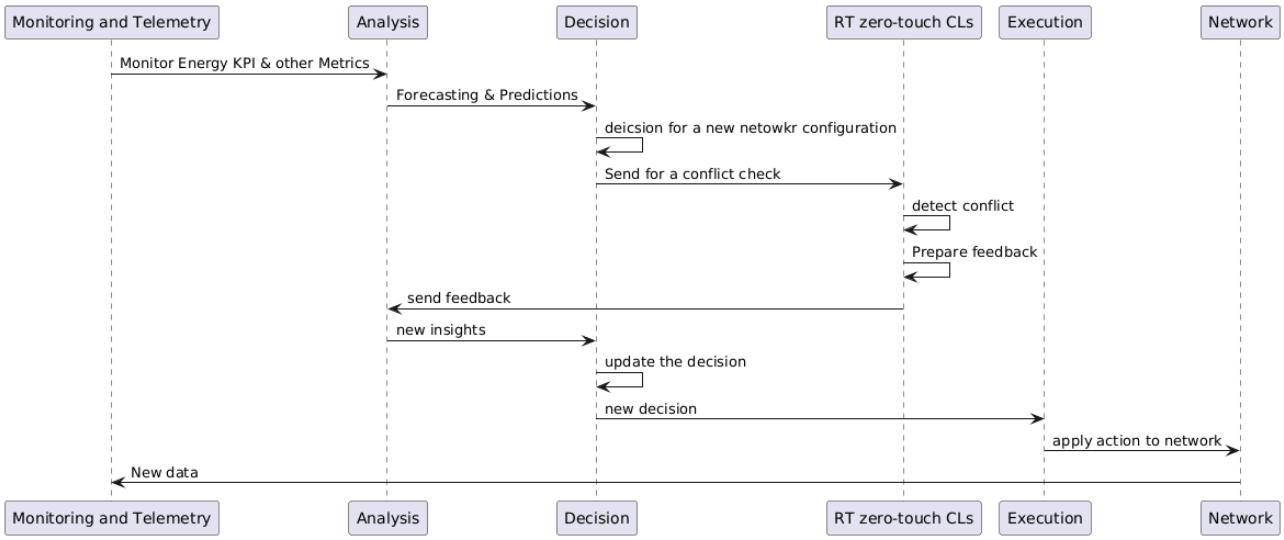

Figure 2-67: Workflow of ML-based network configuration selection for energy saving

The following steps explain the workflow:

Step 1: Monitoring and Telemetry functionality can be used to monitor the total cell energy consumption and other metrics such as the cell throughput, latency, RRC and E-UTRAN Radio Access Bearer (ERAB) success rates.

Step 2: CL analysis module utilises a ML-based forecasting with looking at the trend in a horizon. More specifically, the energy forecast is done per cell for a day in advance.

Step 3: Those insights can be then sent to the Decision module where the CL can propose a new network configuration. With the proposed algorithm, it decides whether make a capacity cell sleep or not. However, this decision is not directly applied to the network, since a conflict check is needed, so this request can be sent to the RT zero-touch CLs functionality, where the CL coordinator component can apply the conflict detection mechanisms.

Step 4: If a conflict is detected with other network KPI, the RT zero-touch CL automation functionality can send feedback indicating this conflict, so that the CL can updates its action. The conflict is checked with other KPIs such as throughput, RRC and ERAB success rate.

Step 5: A new network configuration with no or minor conflict can be then applied to the network. In this case, a cell can be either put in sleep mode or not.

Step 6: The Monitoring and Telemetry functionality can collect new data, so that the impact of the final action can be monitored.

Evaluation

At initial test and implementation has been performed, demonstrating a 10-12% reduction in energy consumption across pilot sites is achieved. This energy efficiency improvement is accomplished without any compromise on the overall network performance, it is confirmed that no degradation in critical parameters such as RRC success rates, ERAB success rates, or call drop rates across all frequency bands is observed. Moreover, the stability and reliability of key operational metrics is maintained. Traffic volume remains consistent with expected patterns, mobility success rates are maintained at optimal levels, and latency metrics show no adverse fluctuations.

In addition to these accomplishments, the network throughput remains steady, with both downlink and uplink performance aligning seamlessly with historical benchmarks. This outcome not only demonstrates the ability to implement energy-saving measures effectively but also highlights the robustness of the network's core KPIs under optimised conditions.

2.2.2.5 Resource assignment for federated learning

Description

This implementation is based on the multi-domain federated learning component of the smart management framework. It extends the capabilities of the service and resource management functions as described in [28.538] and depicted in Figure 2-68 enabling the network to efficiently allocate compute resources for federated learning. It therefore facilitates network services provisioning and compute resources assurance and configuration by directing the orchestration of resources to deliver on training services that constitute Compute as a Service use cases, foreseen as a growth market in 6G given the considerable number of connected verticals envisaged.

The Multi-domain Federated Learning algorithm in the framework can work closely with the Monitoring and Telemetry component to manage the life cycle of model training services. The lifecycle of the multi domain federated learning is shown in Figure 2-68. The *Service Management* receives a service request for training a ML model and redirect it to the *Multi-domain federated learning* algorithm. The latter requests for information regarding the data sources, computing resources and network links – jointly termed *data resources*- from the resource management function. Using this information, the algorithm searches for the configuration of data resources that minimises energy consumption. It then informs the *resource management* function of the optimal selection which then provisions these data resources to deliver the federated learning service.

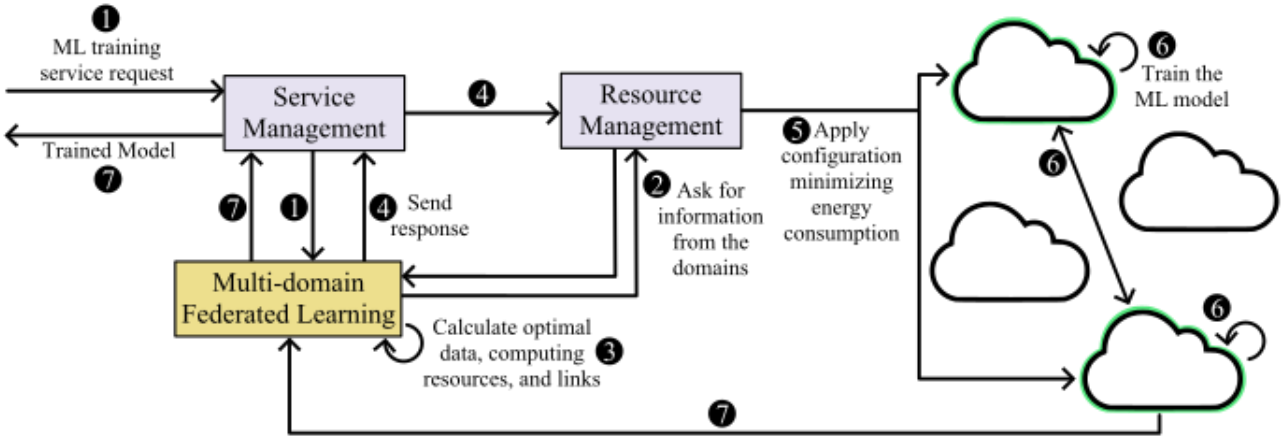

Figure 2-68: Lifecycle of the federated learning service on edge resources.

Workflow

The sequence diagram of the resource assignment for federated learning is shown in Figure 2-69. The process begins when the client submits a service request to the Service Management, which then redirects it to the multi-domain federated learning algorithm. The latter requests specific details about the *data resources* from Resource Management. With this information, the resource management function provisions the *data resources*, optimizing for minimal energy consumption while ensuring the required service level objectives (such as accuracy) are met. Once the resources are provisioned, the model training begins. This phase involves a cycle of data updates from data sources to computing nodes, local model training at each node, and weight aggregation across nodes based on the defined communication pattern. After the model reaches its final version, it is sent back to the client as depicted.

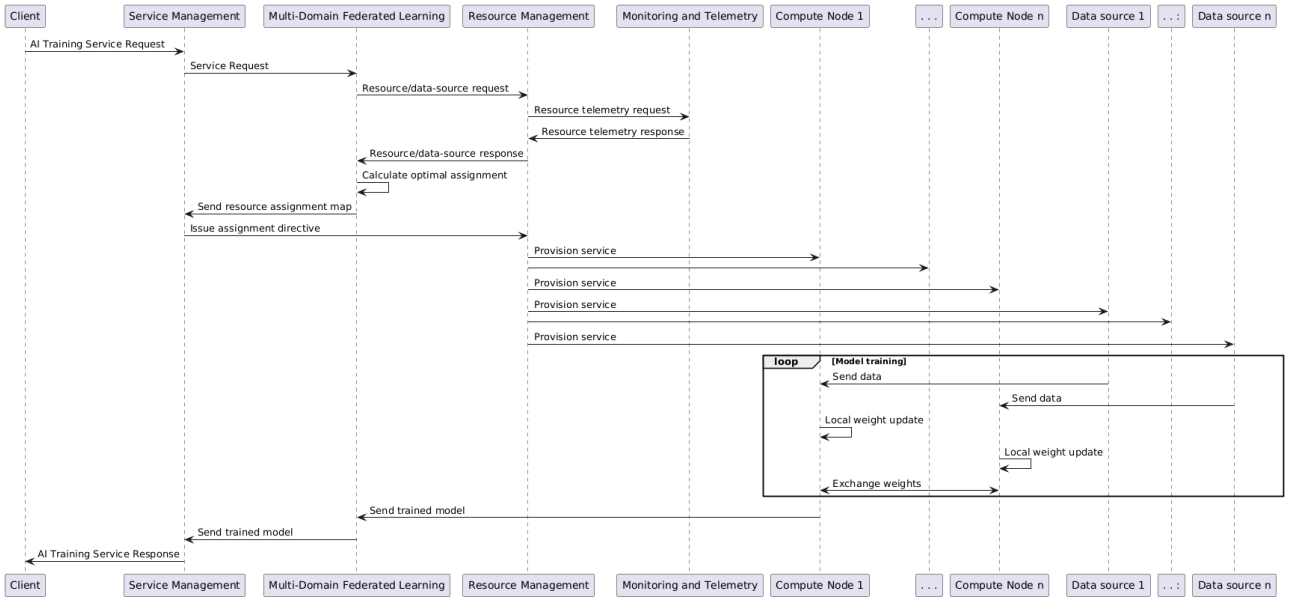

Figure 2-69: Workflow of Multi-domain federated learning.

Evaluation

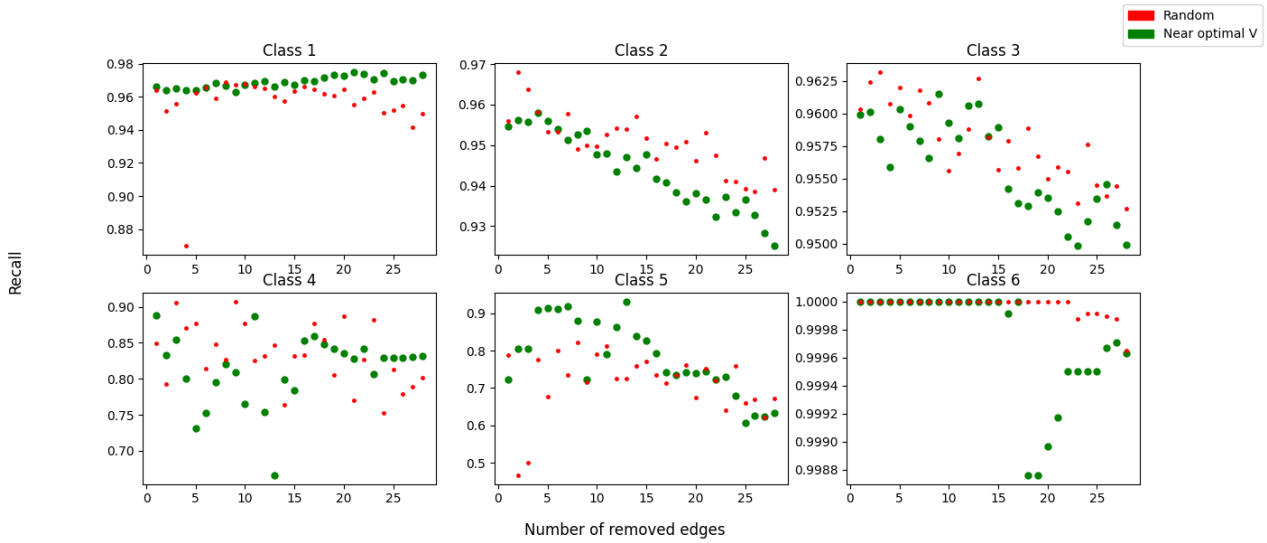

Figure 2-70: Model accuracy variation with optimised data resources.

The algorithm minimises (prunes) the data resources engaged in training. The V distance compares datasets by considering their size and the type of data they contain. Without a loss of generality, the evaluation was carried out on a testbed with 9 domains each with a data source and one compute resource. Levering this metric the algorithm reduced the number of interactions among nodes in the network considerably. The reduction consists of 30% fewer hops per epoch between learning peers and 10% fewer epochs resulting in an overall

reduction of 37% without compromising the quality of the eventual aggregated model – specified by the recall [STR15] - as shown in Figure 2-70. This reduction translates into a significant decrease in the energy consumed by the AI training process. Note that the concrete values in energy savings achieved are dependent on the actual implementation of the physical network given that different media, such as fiber links or radio links, have different energy requirements.

2.2.2.6 Flow Reconfiguration via Dynamic Monitoring and Closed CLs in Deterministic Networks

In deterministic networking, ensuring real-time performance and reliability is critical for time-sensitive flows. The implementation described here focuses on flow reconfiguration based on dynamic monitoring and closed control loops. In this implementation, monitoring data is used to continuously evaluate the performance of flows and react to issues, such as unusually high E2E delays due to e.g., routing misconfigurations, misbehaving flows, and hardware errors. The primary objective is to create a real-time, zero-touch management framework that autonomously monitors network performance and dynamically adjusts flows in response to observed conditions, without requiring human intervention.

In-band network telemetry (INT) provides an ideal monitoring technique for collecting fine-grained, per-flow data. However, the addition of telemetry information inside the real-time data stream results in a significant overhead. This means that less "real" data can be embedded inside each packet, resulting in a performance degradation. This is especially the case if telemetry data, such as queueing time, is collected in a hop-by-hop fashion, such that the data packet grows at each hop. Moreover, collecting a great amount of per-flow INT data by default results in significant overhead for both collecting the data (bandwidth) and analysing the collected historical data (computation time). The goal of this implementation is to minimise the overhead caused by INT, while providing a reasonably accurate view of the network-wide performance of flows, minimising the analysis and memory required by the control loop, and providing adequate reaction time to faulty network behaviour. It aims to do so by exploring several trade-offs between INT, active telemetry, and passive telemetry methods. Moreover, this implementation explores sketching methods for scalable and memory efficient data analysis.

This implementation combines three components of the smart management framework:

1. The Network Programmability system. This component enables routing a time-sensitive traffic flow in a deterministic network.
2. The Monitoring and Telemetry functionality. This component is responsible for configuring the monitoring strategies in the network (such as INT) as well as the collection of INT and per-hop data (e.g., using active or passive telemetry).
3. The Real-time zero-touch control loops automation and coordination functionality. This component provides the real-time analysis of collected monitoring data and executes fitting actions to reconfigure the monitoring strategy to gain more insights or reconfigure the path of the flow itself.

Figure 2-71 shows the basic interactions between these components (framework components are shown as rectangular blocks). As it can be appreciated:

- The SourceApp, which is an application that acts as the source of a deterministic traffic flow (e,g,, a factory robot), requests a flow through the Network Programmability system, which reserves resources in the Network Infrastructure and configures the Monitoring and Telemetry functionality. SourceApp then sends traffic to DestinationApp (the application consuming the traffic from the SourceApp, such as a control server) over the Network Infrastructure.
- The Monitoring and Telemetry functionality configures INT on the Network Infrastructure as needed and continuously receives data from the infrastructure.
- The Monitoring and Telemetry functionality feeds data to the CL, which detects issues and sends reconfiguration commands back.
- The CL requests a flow reconfiguration from the Network Programmability system, which then reserves resources in the Network Infrastructure as needed for the flow.

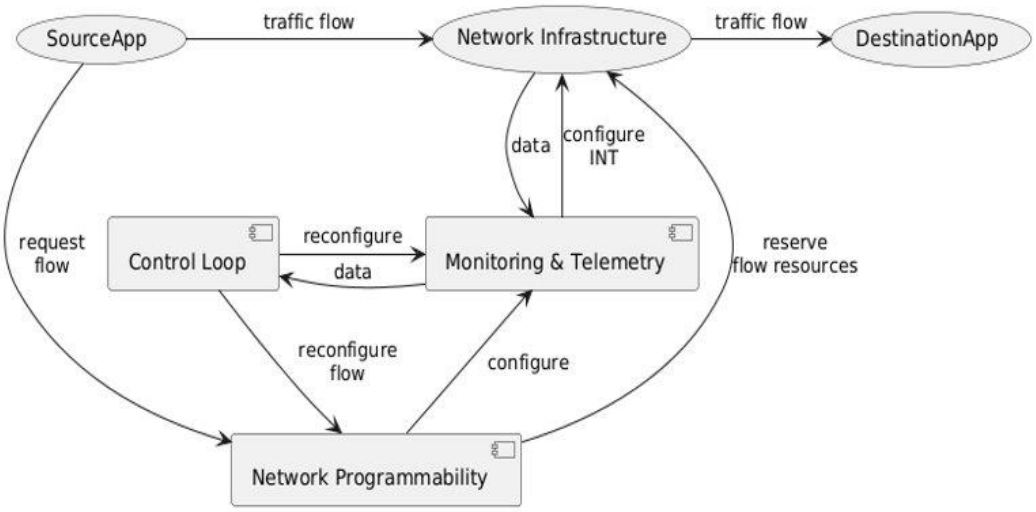

Figure 2-71: Interactions of the CL for dynamically monitoring and rerouting deterministic flows.

Workflow

The process, as shown in Figure 2-72, begins with a flow request from the source application (SourceApp) that is handled by the network programmability (controller) component, which takes care of routing and reserving the necessary network resources. To ensure minimal overhead in the network, lightweight in-band network telemetry (INT) is initially applied to the flow, enabling only E2E delay measurements. For the actual data packet, this only requires INT headers and the ingress timestamp to be added at the first hop. Other hops don't need to modify the packet. This telemetry data is periodically reported to the closed control loop system, which continuously analyses it to detect anomalies.

When a significant issue, such as an unusually high E2E delay, is detected, the closed control loop escalates the monitoring by enabling fine-grained INT, such as path tracing, to ensure the intended route is taken. This deeper level of telemetry provides more detailed insights, which are then analysed to identify the root cause, such as a routing misconfiguration. If necessary, the closed control loop system triggers a reroute of the flow, ensuring that the network maintains its deterministic performance standards. After a reroute is triggered, the choice can be made to immediately resort to light-weight INT again, or to keep the additional measures for some time until it is established that everything goes fine again.

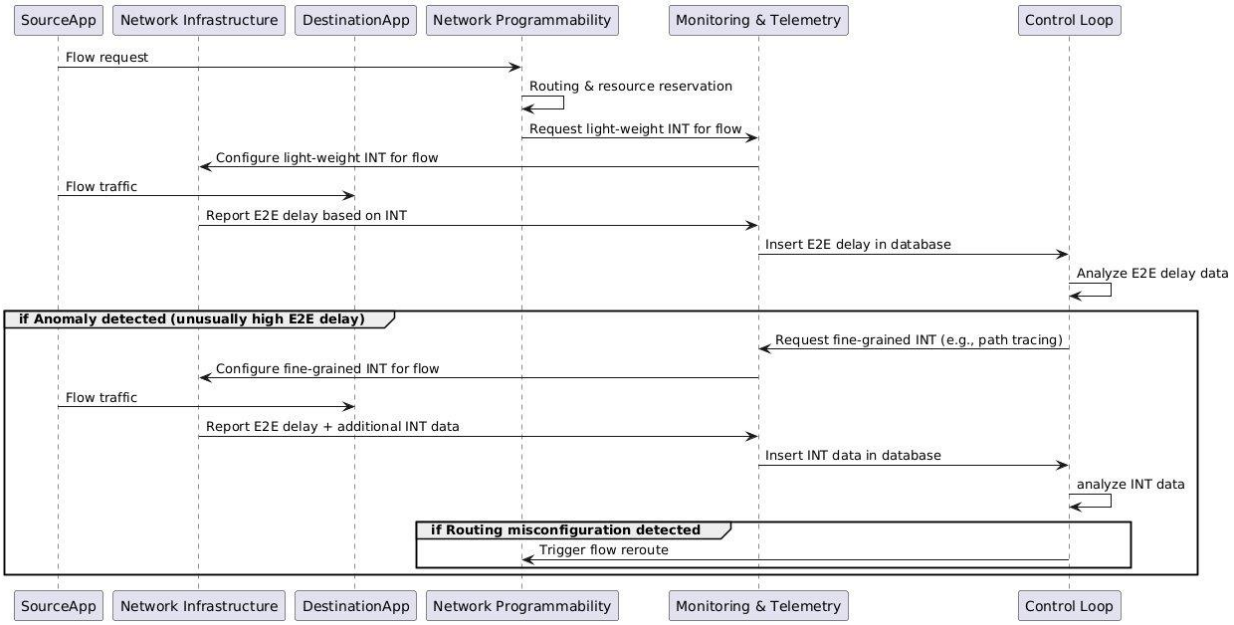

Figure 2-72: Sequence diagram in the scenario of an anomaly in the deterministic network.

Evaluation

This implementation has been evaluated on a testbed, where a time-sensitive flow (sending data at a rate of 100 kbit/s) was established and monitored. This experiment was conducted twice: once using INT-MD and once using INT-MX mode. During its runtime, an anomaly was introduced, which caused high E2E delays. As seen in Figure 2-73, the CL succeeds in detecting the unusually high E2E delay and then reduces this delay back to acceptable levels. The bandwidth overhead due to INT is depicted in Figure 2-74. Here, a spike in INT bandwidth can be seen around the same time that the delay spike happened. This is because at first, only E2E delays are measured using INT. However, after detecting a spike, the CL decided to augment the INT monitoring to pinpoint the issue. Since it now receives richer monitoring data, it can analyse this data and fix the issue; in this case it will trigger a reroute of the flow through an alternative path. This causes the E2E delay returning to normal levels, and after that it resorts again to only monitoring E2E delays, such that the INT bandwidth overhead also returns to normal levels. Finally, Figure 2-75 shows a memory analysis of sketching-based methods versus per-flow data. This implementation explores sketching as a technique for measuring high E2E delays instead of expensive per-flow data. With sketching, it is possible to approximately keep track of the packet counts of a flow, and the frequency of measured E2E delays in a flow, e.g., using two count-min sketches. By combining both sketches, it is possible to detect unusual delays. Although it is an approximation, it can be tuned to a certain accuracy, depending on the use case. The upside is that sketching requires *O(1)* memory, enhancing the scalability of this implementation.\

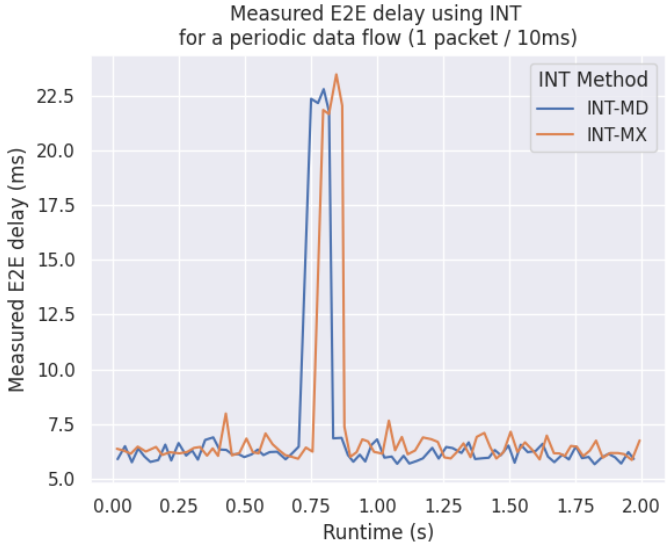

Figure 2-73: CL detects and corrects high E2E delays during flow runtime.

Beyond the lab experiments, this implementation has also been included as part of the project PoC#B.1, demonstrating continuous monitoring and reconfiguration of latency-sensitive flows (e.g., between cluster environments). Finally, this implementation impacts the following KPIs:

- Availability: By mitigating unacceptable E2E delays, the application (service) can continue as expected with the necessary QoS. In the experiments, <2% of all packets were impacted.
- Automation: The time it takes for the CL to complete one loop here depends on the scenario: as the collection of fine-grained telemetry data is done at a network node, this data needs to be sent to the SDN controller for further analysis. This introduces some communication overhead in the CL loop, which can be expected to be at most in the order of milliseconds.
- OPEX: The CL operates fully autonomously, requiring no human intervention. It also reduces the resources required to monitor and analyse traffic flows, as depicted in Fig. 2.74 and 2.75.
- Latency: The latency of the traffic flow is reduced a runtime to acceptable levels after high E2E delay detection.
- Reliability: As shown in Fig. 2.73, the CL successfully detects and corrects the high E2E delays. The recovery time here is around 120 ms (~4 packets impacted), which constitutes the period after the first detection of the high E2E delay, until recovery to normal levels.

- Scalability: By applying dynamic monitoring, the bandwidth overhead for INT is reduced. As shown in Fig. 2.75, the average INT bandwidth overhead is kept lower than 10 kbit/s on average. This is significantly lower than the peak overhead when more fine-grained INT is needed. Moreover, the CL requires only *O(1)* memory complexity for detecting high E2E delays. This is more scalable than keeping per-flow state which scales linearly with the number of flows, as shown in Figure 2-75.

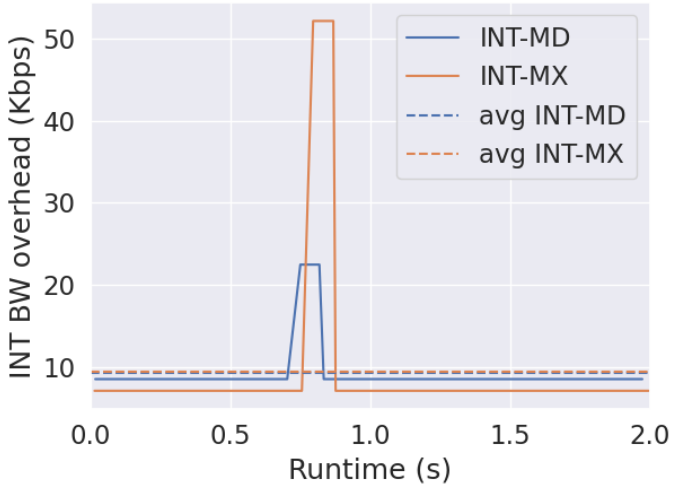

Figure 2-74: INT bandwidth overhead during flow runtime (increase during high E2E delays).

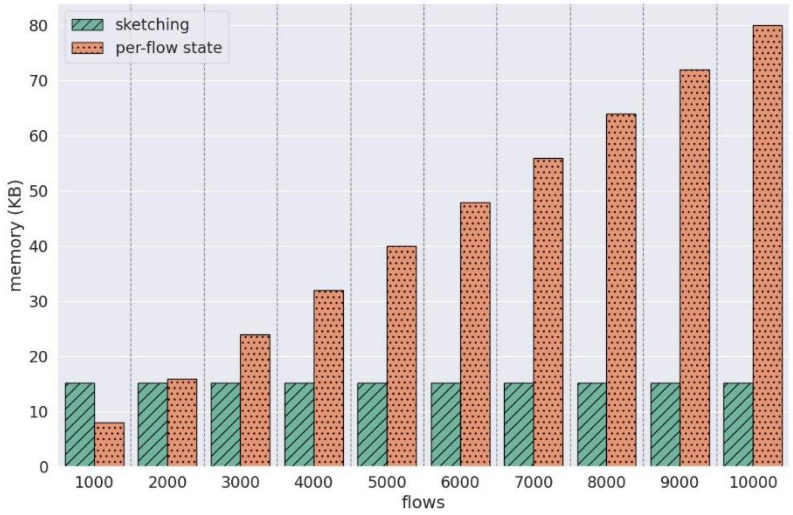

Figure 2-75: Memory analysis of sketching vs per-flow state for measuring E2E delay.

2.2.2.7 Edge convergence over federated resources for the network continuum

This implementation targets to explore the capabilities of the CAMARA EdgeCloud APIs [CEC24] in the management of the compute resources in the network continuum, and the possibility to extend them to be used with federated resources of external administrative domains. Developed as a testbed, the federated resources are available within a DLT network, and the implementation extends the EdgeCloud APIs for access to federated resources in a seamless way with minimal changes in the APIs.

The implementation relies on the SLA-driven Federated Orchestration system, the Multi-agent System for Multi-cluster Orchestration (related with the Multi-Cluster Resource Manager), and the MCE functionality in the management framework.

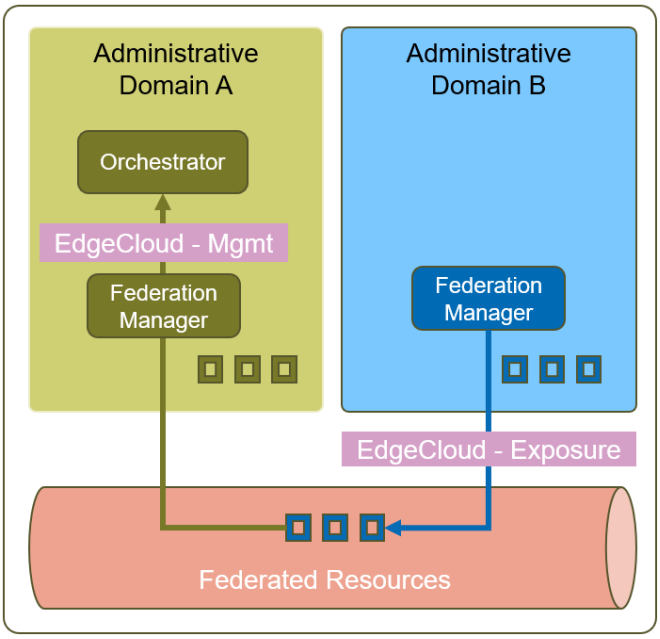

Figure 2-76: Federated resources shared between different administrative domains.

Figure 2-76 shows the testbed configured for this implementation, where a DLT network with two administrative domains was created, as well as the smart contracts associated to an agreement established among parties after a defined negotiation, in the following way:

- One of the administrative domains exposes its federated resources (the small squares) available for sharing with other domains over the DLT network.
- The other administrative domain needs resources to fulfil a service. These resources could be their own resources or external ones, i.e., including network continuum resources by including those resources exposed in the DLT by the other administrative domain. If the external resources were required, the service provisioning area would request for an agreement (which would need to be created if it did not exist).
- Using the updated EdgeCloud APIs, the continuum among partners would be extended, being an agile method for increasing the resources capability.

Figure 2-77 illustrates the workflow followed in a complex multi-domain environment, where multiple providers (A and B) and their respective orchestrators collaborate to deliver services:

- Service provider B exposes its available resources to the federation through the Management Capabilities Exposure functionality.
- The multi agent system for multi cluster orchestration system is responsible for the service orchestration. To do that, the service provider A requests a service instance to the orchestrator of its domain.
- Orchestrator A needs resources from third parties to perform the design of the service instantiation:
    - It gets the available resources from the federated ones for a specific region or zone where the service will perform its activity.
    - With the use of the Federated Orchestration enabler, it orchestrates the service (agreement negotiation and token request included).
- Finally, the service is ready to be used by the user from Domain A.

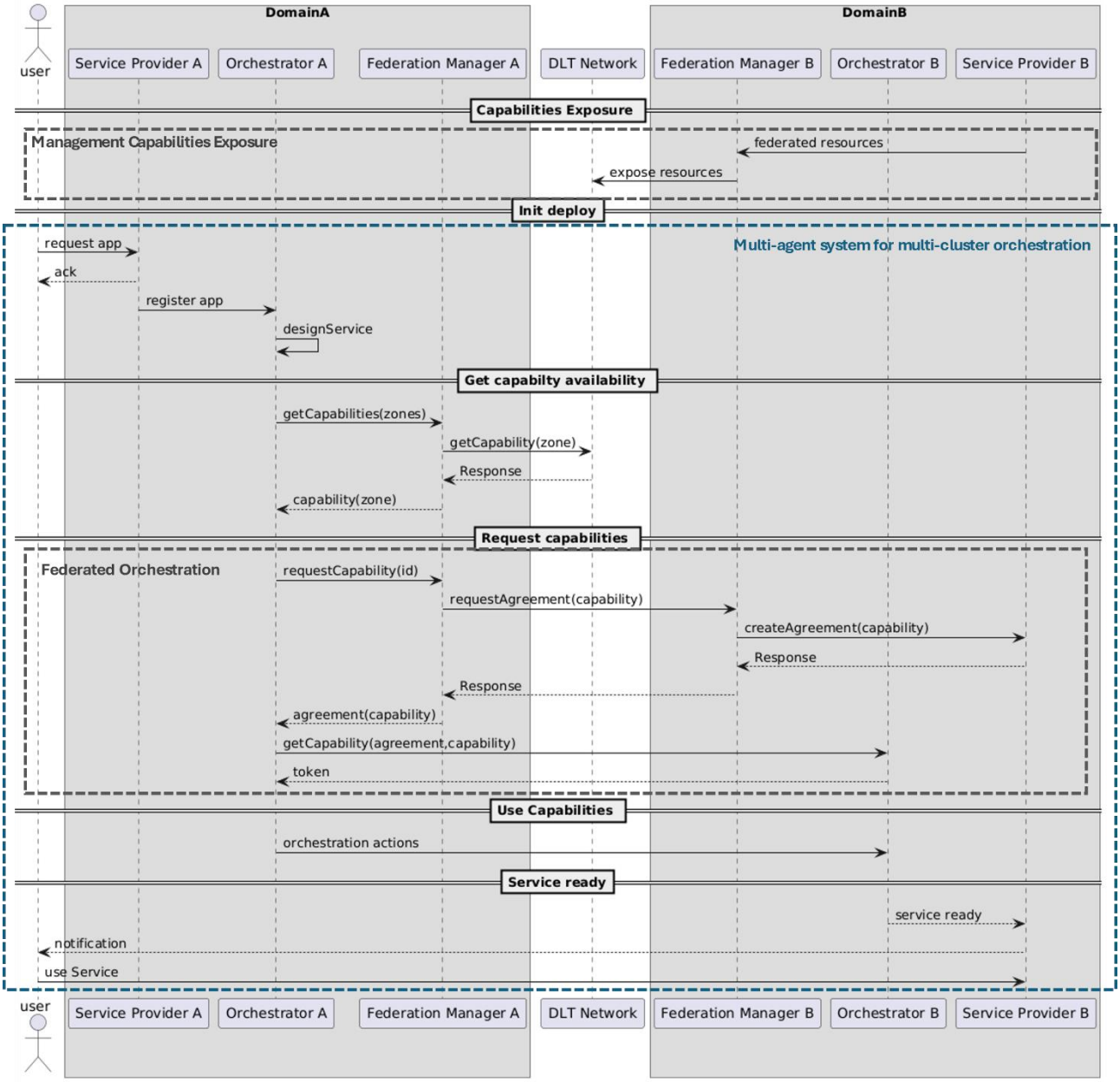

Figure 2-77: Federated resources from different administrative domains.

Evaluation results

The CAMARA API EdgeCloud definitions were validated as effective for federated exposure and management of compute capabilities. These definitions successfully considered service availability zones, ensuring harmonisation of capabilities across different environments.

Smart contract agreements were effectively integrated using TMForum Agreement definitions within orchestrator actions. This integration, managed by business or end-to-end (E2E) layers, employed Distributed Ledger Technology (DLT) and the CAMARA's Identity and Consent Management API for enhanced governance and security.

Modifications were identified as necessary in the CAMARA APIs to support the discovery, management, and deployment of services in a federated environment. This highlighted areas where APIs needed to evolve for better resource federation.

A common DLT was proposed to enable secure, seamless integration across multiple stakeholders. This would simplify operations in complex environments and promote easier usage across the ecosystem.

The interactions in the continuum were extended using resources from different domains and simplifying operations in multi-cluster orchestration scenarios. This was achieved through coordinated actions, ensuring smoother operation across multiple stakeholders and clusters.

Regarding the KPIs addressed, the following can be highlighted:

- Scalability: regarding the usage of resources from a larger catalogue (combination of own resources and federated ones), using service components from several providers and locations.
- Latency: with the use of resources with better capabilities and closer to the user. The use of federated resources in the nearby areas (device to access network) resulted in 12 ms (measurements from in-house testbed), compared to the 40 ms measured for the core domain resources, and the 60 ms for hyperscalers.
- Reliability: easing service migration between domains, facilitating new paths when the defined service is not working.
- Availability: being able to create services with better features to comply with the personalised SLAs of client services. Also, the time between adding a federated resource in the network and the time for connecting it to the federated resource management was less than 1 minute (Figure 2-78).
- Automation: with the use of external capabilities in an agnostic way (a service provider could use external resources in a similar way as their own resources).

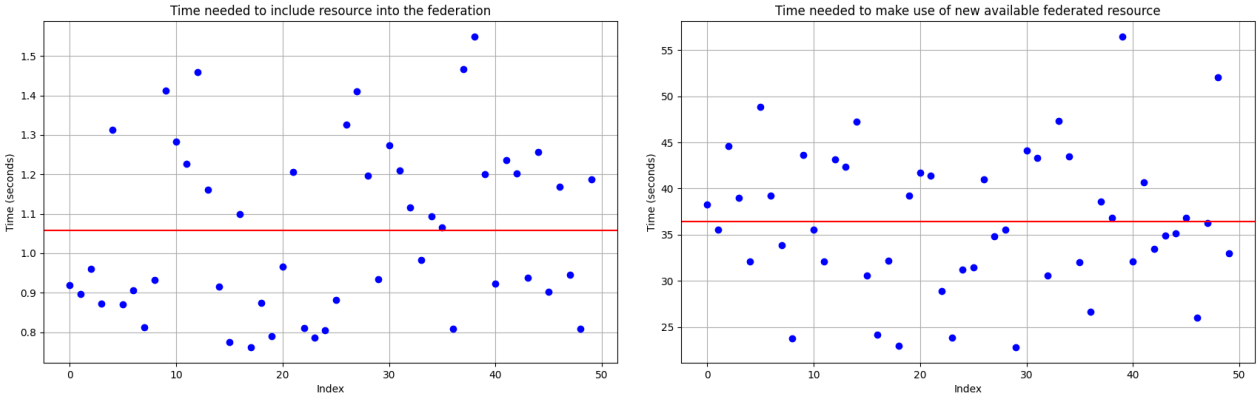

Figure 2-78: Times to add and to use a federated resource from testbed.

The testbed evaluated the impact of utilizing external resources, particularly focusing on the time required to link these resources and the associated entities. It also considered the necessary information for this process. The results from this testbed are shared with the CAMARA project for further evaluation, offering valuable insights for future improvements.

2.2.3 Contributions to the software developer's community

2.2.3.1 Open Software releases

The following table shows the contributions to the Open-Source community done from this WP6 with their main features and the link to the public repositories where they are available.

Table 2-1: Open-source Software Releases

Software component	SW Origin	Licence	Available in	Related Component/ UseCase	PoC / Demo
ETSI TeraFlowSDN	Existing SW - Enhanced in project	Apache 2.0	TeraFlowSDN Website	Network Programmability	PoC#B.1
Digital Ledger Technology (DLT) for service federation	New SW - Created in project	Apache 2.0	GitLab	Federated orchestration system	PoC#B.1
Infrastructure Layer Emulator (ILE)	New SW - Created in project	Apache 2.0	GitHub	Decentralised Orchestration System	PoC#B.1

HX MLOps	New SW - Created in project	Apache 2.0	GitHub	Sustainable MLOps	Testbed
Routing protocols for deterministic networks	Existing SW - Enhanced in project	Apache 2.0	GitHub	Network Programmability	Testbed
6G latency sensitive service for Smart Manufacturing	New SW - Created in project	Apache 2.0	GitLab	Overall M&O System enablers.	PoC#B.1

In the following, additional details are provided regarding each of these components.

TeraFlowSDN

TeraFlowSDN (TFS) is a cloud-native, software-defined networking (SDN) controller designed to manage and orchestrate transport networks across multiple technological domains. TeraFlowSDN also facilitates modular network architectures with multiple centralised controllers and synergises with services like MEC BandWidth Management to optimise network resource allocation, aligning with standards from ETSI, IETF, and TM Forum to promote automation, flexibility, and technology-agnostic network management.

Hexa-X-II has contributed to the following novel TFS features:

- Replace DLT Gateway functionality with an opensource and Hyper Ledger v2.4+ compliant version.
- Implement SBI Driver for Nokia SR Linux L2 VPNs through gNMI
- Extend gNMI-OpenConfig SBI driver
- Implement ETSI MEC Bandwidth Management API in NBI component

More details can be found at: [TFS24]

Digital Ledger Technology (DLT) for service federation

This solution provides an SLA creation and policing scheme aimed at the provisioning of services beyond the domain of a network operator. Given the expected disaggregation of the provider landscape and the emergence of many mobile virtual network operators, services initiated by one can be extended or migrated to another to ensure continuity and improved QoE. This contribution provides a mechanism through this service federation can be executed, monitored and billed.

The following features of this solution have been developed in the context of Hexa-X-II:

- Development of the East/West Bound API following the Open API specification to ease integration to generic orchestrators.
- Refactoring of the codebase to include support for various M&O solutions beyond OSM e.g. Kubernetes.
- Creation of smart contracts corresponding to an industrial robot use case.

Infrastructure Layer Emulator (ILE)

The Infrastructure Layer Emulator (ILE) is a software designed to replicate key features of the infrastructure envisioned for future 6G networks. Specifically, it enables:

- The simulation of extensive computing node deployments, since the 6G M&O systems are expected to operate over a vast and distributed network continuum, in a cloud-native scale.
- The integration of diverse types of computing nodes. The network continuum is envisaged not only large but also inherently heterogeneous, comprising various types of nodes.
- The emulation of multiple stakeholders. It supports testing scenarios that can involve different network operators and other stakeholders.
- The representation of diverse network domains, including core, edge, and extreme-edge.
- The emulation of the dynamic behaviour at the extreme edge. The emulator accounts for the high volatility of extreme-edge resources, such as devices that frequently connect or disconnect and exhibit changes in properties like memory usage, CPU load, or battery level in the case of battery-powered devices.
- The deployment of realistic network services. The emulator can support deploying and testing actual network services in a simulated network environment.

Software resources like the ILE are considered necessary to creating realistic testing environments for evaluating and demonstrating 6G design concepts.

The ILE is implemented using LXD [LXD], a system container and virtual machine manager. LXD provides a unified platform for running and managing full Linux systems in containers or virtual machines. It supports a wide range of Linux distributions, from lightweight variants suitable for deploying many nodes on small-scale equipment to complete distributions capable of hosting complex network services. LXD can scale from a single instance on one machine to a cluster simulating a full data centre, making it versatile for a range of scenarios. This setup can enable the deployment of different orchestration resources and mechanisms (e.g., OSM or Kubernetes) on these containers, and subsequently facilitates the deployment of network services on top of the orchestration layer. Also, specific configuration parameters allow users to define the behaviour of the emulator (e.g., connection/disconnection patterns on the extreme-edge nodes).

Additionally, visualisation is provided through a graphical user interface (GUI) that displays the different stakeholders and network domains across the continuum (Core, Edge, and Extreme Edge) and the devices belonging to them. Figure 2-79 shows a screenshot of this GUI, illustrating a setup with three Mobile Network Operators (MNOs). Each MNO is represented by the white polygons with core and edge resources (depicted as circles and diamonds, respectively), as well as extreme-edge nodes beyond their domains. Green and red circles represent real containerised Linux machines capable of communication and service hosting, with their colour indicating connection status (green for connected and red for disconnected). In the live setup, the extreme-edge nodes are commonly "blinking" (asynchronously changing from green to red or vice versa) as their availability changes. Additionally, the figure represents three deployed services (thick lines) connecting specific nodes across infrastructure resources within and beyond the MNO domains.

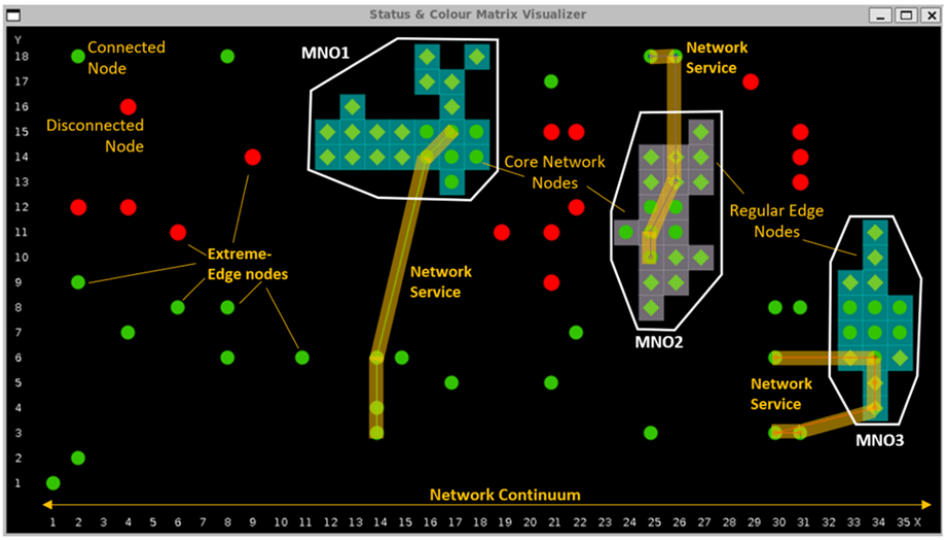

Figure 2-79: Infrastructure Layer Emulator GUI.

HX MLOps

The HX-MLOps software is intended to automate, monitor, and optimise the lifecycle of AI/ML-based network services relying on the well-known DevOps principles for the continuous development and integration (CI/CD) of AI/ML models, but considering the multi-stakeholder ecosystem of the telco-grade environment. It integrates a Python-based command-line interface (CLI) for the users (network operators, vertical industries, network service developers, …) to customise and deploy AI/ML-related workflows tailored to their domains, and a graphical user interface (GUI) for visualizing energy consumption across the workflow's stages and components. The tool also allows to measure energy consumption and the carbon emissions associated to the different stages of the workflows, so enabling users to assess and optimise the environmental impact of the training, deployment, and exploitation of AI/ML-based assets. This HX-MLOps tool supports modular deployments with categorised components, including storage, ML lifecycle management components, serving platforms, observability tools, and energy measurement resources. It also includes APIs for sharing models and datasets among the different stakeholders participating in the development and/or the exploitation of the AI/ML models, fostering collaboration and integration across stakeholders.

Routing protocols for deterministic networks

The implementation of several routing protocols for deterministic networks has been released as open-source software in the context of Hexa-X-II. These include the OMNeT++ implementation of three novel routing protocols: a link-state routing protocol and two exploration-based routing protocols. These implementations can be used to test, demonstrate, and evaluate the routing protocols, and they can also be extended for further research.

Within Hexa-X-II, these software implementations were further enhanced and evaluated in large-scale scenarios. Comprehensive evaluations were conducted to assess their performance and scalability under various conditions to assess their applicability in real-world deterministic networking environments [MCP+24].

6G latency sensitive service for Smart Manufacturing

An open-source smart manufacturing application was developed for managing the operation of a static robotic arm, composed of two main service chains: one regarding the teleoperation of a robotic vehicle performing surveillance of the workspace around the arm and a second regarding the generation of alerts to safely pause and start the operation based on an ML-based object detection component analysing the video stream of the vehicle.

2.2.3.2 OpenAPIs

The following table shows the OpenAPIs generated in this WP6 together with their main features and the link to the public repositories where they are available.

Table 2-2: Open APIs Specification

API	SW Origin	Licence	Available in	Related Component / UseCase	PoC / Demo
DLT Service Federation Open API	New - Created in project	CC Attr. 4.0 International	Zenodo	Federated Orchestration	PoC#B.1
Sustainable MLOps models sharing API	New - Created in project	Apache 2.0	Zenodo	Sustainable MLOps	Testbed
TMF640 Service Activation Management API	Existing - Enhanced in project	Apache 2.0	GitHub Zenodo	Network programmability	Testbed
TMF644 Resource Function Activation Management API	Existing - Enhanced in project	Apache 2.0	GitHub Zenodo	Network programmability	Testbed
Conflict Detection for CLs Automation and Management API	New - Created in project	CC Attr. 4.0 International	Zenodo	Real-time zero-touch control loops automation and coordination functionality	Testbed
MEC Bandwidth Management (MEC 014)	Existing	BSD-3-Clause	Github	Network programmability	Testbed
CAMARA QoD	Existing	Apache 2.0	Github	Network programmability	Testbed
Monitoring jobs configuration	Existing – Enhanced in project	Apache 2.0	Zenodo	Monitoring and Telemetry	PoC#B.1
Query historical monitoring data	Existing	MIT	InfluxDB site	Monitoring and Telemetry	PoC#B.1
Multi-cluster extreme-edge resource orchestration API	New - Created in project	Apache 2.0	Zenodo	Multi-agent system for multi-cluster orchestration	PoC#B.1

Closed Loop Governance - Catalogue API	New - Created in project	Apache 2.0	Zenodo	Real-time zero-touch control loops automation and coordination functionality	PoC#B.1
Closed Loop Governance – Lifecycle Management API	New - Created in project	Apache 2.0	Zenodo	Real-time zero-touch control loops automation and coordination functionality	PoC#B.1
MEC exposure and experience management API	Existing-Modified in project	CC Attr. 4.0 International	Zenodo	Edge convergence over federated resources for the computing continuum	Testbed
Sustainable MLOps Workflow Info Collector API	New – Created in project	Apache 2.0	Zenodo	Sustainable MLOps	Testbed
Security API	New - Created in project	CC Attr. 4.0 International	Zenodo	Management Capabilities Exposure	Testbed
Subscription API	New - Created in project	CC Attr. 4.0 International	Zenodo	Management Capabilities Exposure	Testbed
Listing API	New - Created in project	CC Attr. 4.0 International	Zenodo	Management Capabilities Exposure	Testbed
Trust Management API	New - Created in project	GPL-3.0	Zenodo	Trust Management System	Testbed
Trust Evaluation Function API	New - Created in project	GPL-3.0	Zenodo	Trust Management System	PoC#A.1
LoTAF API	New - Created in project	GPL-3.0	GitHub Zenodo	Trust Management System	Testbed
DLT Service Federation Open API	New - Created in project	CC Attr. 4.0 International	Zenodo	Federated Orchestration	Testbed

2.2.4 Supporting technologies

The following tables show the main technologies that support the technical enablers of the smart management framework, indicating their origin and, where applicable, whether they have been developed specifically in the context of the project. The information is organised according to the different scopes of in the framework.

Table 2-3: Whole network continuum scope – Overall M&O Solutions

Component	Supporting Technologies	Source
Multi-agent system for multi-cluster orchestration solution	Service orchestration mechanisms – Joint Scaling and Placement	Project scope - https://gitlab.com/netmode/hexa-service-orchestration
	Kubernetes - Resource Management Platform	External - https://kubernetes.io/
	Karmada – Multi-cluster orchestration platform	External - https://karmada.io/
	Prometheus – Monitoring Engine	External - https://prometheus.io/
	Network Service Mesh	External - https://networkservicemesh.io
Decentralised Orchestration solution	LXD - Open-source solution for managing virtual machines and system containers.	External - https://canonical.com/lxd

	ILE – Infrastructure Layer Emulator.	Project scope - https://gitlab.com/decentralized-continuum-orchestration/infrastructure-layer-emulator
	Node JS – Open-source JavaScript runtime environment.	External - https://nodejs.org
	TimescaleDB – Database based on PostgreSQL suitable for managing timeseries data.	External - https://www.timescale.com
	RabbitMQ – Open-source message-broker solution.	External - https://www.rabbitmq.com
	PyTorch – Deep Learning framework to build end to end machine learning workflows.	External - https://pytorch.org
	Flask – Web Server Gateway Interface (WSGI) web application framework.	External - https://flask.palletsprojects.com

Table 2-4: Whole network continuum scope – Specific Functionalities

Component	**Supporting Technologies**	**Source**
Monitoring and Telemetry	P4 - A domain-specific open-source language for network devices, specifying how data plane devices (switches, NICs, routers, filters, etc.) process packets.	External - https://p4.org
	OpenTelemetry - High-quality, ubiquitous, and portable telemetry to enable effective observability	External - https://opentelemetry.io
	Apache Kafka – open-source distributed event streaming platform	External - https://kafka.apache.org/
	Prometheus – open-source monitoring system	External - https://prometheus.io/
Real-time Zero-touch CLs automation and coordination	Apache Kafka – open-source distributed event streaming platform	External - https://kafka.apache.org/
	Telegraf – open-source agent for data collection	External - https://www.influxdata.com/time-series-platform/telegraf/
	Prometheus – open-source monitoring system	External - https://prometheus.io/
	InfluxDB Open Source – time-series database	External - https://www.influxdata.com/products/influxdb/
	FastAPI - web framework for building APIs with Python	External – https://fastapi.tiangolo.com/
	SQLAlchemy - Python SQL Toolkit and Object Relational Mapper	External – https://www.sqlalchemy.org/

Management Capabilities Exposure	Docker: Accelerated Container Application Development	External - https://www.docker.com/
	FastAPI - web framework for building APIs with Python	External – https://fastapi.tiangolo.com/
	Redis – The Real-time Data plataform	External - https://redis.io/
	Apache Kafka – open source distributed event streaming platform	External - https://kafka.apache.org/
SLA-driven Federated Orchestration	Kubernetes (K8s) – Container orchestration platform	External - https://kubernetes.io/
	Helm charts – K8s package deployment manager	External - https://helm.sh/
	Go-Ethereum – implementation of Ethereum Blockchain protocol supporting smart contracts	External - https://geth.ethereum.org/
	Smart contract based blockchain service federation	Project scope - https://gitlab.netcom.it.uc3m.es/hexa-x-ii/dlt-federation
Trust Management	Docker: Accelerated Container Application Development	External - https://www.docker.com/
	FastAPI - web framework for building APIs with Python	External – https://fastapi.tiangolo.com/
	Morph-KGC – Engine that constructs RDF knowledge graphs	External - https://morph-kgc.readthedocs.io/en/stable/
	Chowlk Converter – Ontology conceptualisation tool	External - https://chowlk.linkeddata.es/
	Neo4J – Graph Database & Analytics	External - https://neo4j.com/

Table 2-5: Stakeholder's scope – Specific Systems.

Component	**Supporting Technologies**	**Source**
Network Digital Twins Creation Mechanisms	PyTorch – Deep Learning framework to build end to end machine learning workflows.	External - https://pytorch.org
	OMNET++ - Network Simulation program	External - https://omnetpp.org/
	Docker: Accelerated Container Application Development	External - https://www.docker.com/
Sustainable MLOps	Prometheus – Time series DB	External – https://prometheus.io/

	HX-MLops – Hexa-X Machine Learning operators	Project scope - https://github.com/Atos-Research-and-Innovation/HX-MLOps
	Grafana – Observability and dashboarding tool	External – https://grafana.com
	Scaphandre – Intel based energy collectors frameworks	External – https://github.com/hubblo-org/scaphandre
	Kepler – Cloud energy collector framework	External – https://sustainable-computing.io
	Kubernetes (K8s) – Container orchestration platform	External - https://kubernetes.io/
	Helm charts – K8s package deployment manager	External - https://helm.sh/
	FastAPI - web framework for building APIs with Python	External - https://fastapi.tiangolo.com/
	TimescaleDB – Database based on PostgreSQL suitable for managing timeseries data.	External - https://www.timescale.com
	PostgreSQL – SQL Database	External - https://www.postgresql.org/
	TensorFlow Serving – Service to deploy TensorFlow AI models	External - https://www.tensorflow.org/tfx/guide/serving
	TorchServe – Service to deploy Pytorch AI models	External - https://pytorch.org/serve/
	MinIO – a high-performance distributed object storage server	External – https://min.io/
	Kubeflow – Machine Learning toolkit for execute ML pipelines in Kubernetes	External - https://www.kubeflow.org/
Network programmability	ETSI TeraFlowSDN	External – http://tfs.etsi.org
Privacy protection for data analytics	Federated Learning framework along with privacy protection operation (e.g. Homomorphic encryption, Secure Multi-party Computation (SMPC), Differential Privacy, and etc.)	External- http://flower.ai
Secure AI/ML-based control for Intent-based Management	PyTorch – Deep Learning framework to train AI-	External - https://pytorch.org/

	driven agent models and enhance the robustness of the model using adversarial training methods such as FGSM (Fast Gradient Sign Method), and BIM (Basic Iterative Method).	

Table 2-6: Stakeholder's scope – Algorithms.

Component	Supporting Technologies	Source
ML based configuration recommender for energy savings	PyTorch – Deep Learning framework to build end to end machine learning workflows	External - https://pytorch.org
	Docker: Accelerated Container Application Development	External - https://www.docker.com/
Efficient network and service function allocation	PyTorch – Deep Learning framework to build end to end machine learning workflows.	External - https://pytorch.org
	OMNET++ - Network Simulation program	External - https://omnetpp.org/
	Docker: Accelerated Container Application Development	External - https://www.docker.com/
Multi-domain federated learning	PyTorch – Deep Learning framework to build end to end machine learning workflows.	External - https://pytorch.org
	Distributed Training Platform – AI learning testbed that takes as input the network layout, data sources and training scheme to produce a trained model	Project scope - https://gitlab.netcom.it.uc3m.es/hexa-x-ii/distributed-learning-platform
Multi-agent RL for adaptive scaling	PyTorch – Deep Learning framework to build end to end machine learning workflows	External - https://pytorch.org
	OpenAI Gym – open-source Python library for developing and comparing reinforcement learning algorithms	External - https://github.com/openai/gym
Explainability for RL-based control	Gymnasium – open-source Python library for reinforcement learning environments (maintained version of OpenAI Gym)	External - https://gymnasium.farama.org/
	Stable-baseline3 – open-source Python library for reinforcement learning algorithms	External - https://stable-baselines3.readthedocs.io/en/master/

3. Alignment with the Hexa-X-II project work programme

3.1 Interaction with other Work Packages

As introduced in the previous Deliverable D6.3 [HEX224-D63] in what regards its technical approach, WP6 receives inputs from WP2 (E2E System) on the overall platform design and, combining it with information from WP1 (where use cases, values, and requirements are defined) and WP3 (which focuses on the 6G Architecture design), generates outputs back to WP2 in multiple iterations regarding the specific smart network management topics to be integrated into the end-to-end system. Figure 3-1 illustrates this flow of information between WP6 and the other related work packages mentioned.

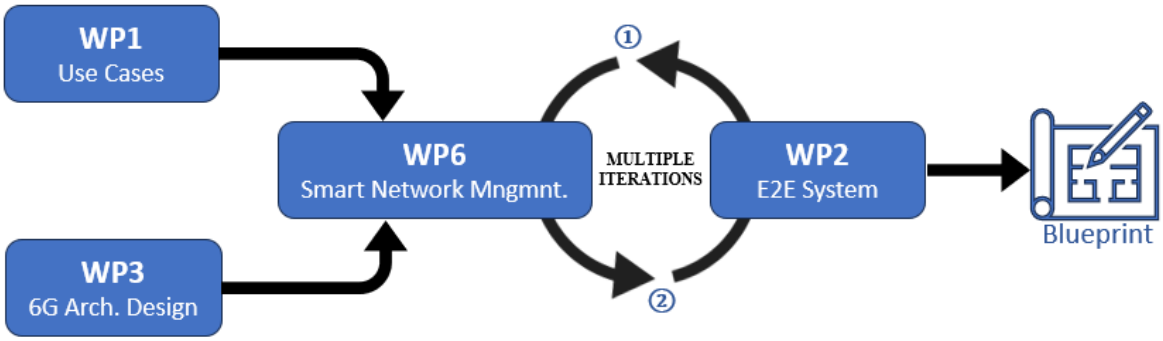

Figure 3-1: WP6 and its relationship with other WPs [HEX224-D63].

In what regards the reporting period covered by this deliverable, these iterations among WP6 and the other mentioned WPs have been mainly the following:

WP1:

- Received updated information on the Use Cases in WP1.
- Specific interactions to align on the procedure to address our work in WP6 regarding the KVIs identification and assessment. The content in Section 3.5 is the result of those interactions.

WP3:

- Consideration of the architectural design concepts carried out in this WP3. Certain components of the smart management framework have been previously defined in WP3 and then incorporated in the WP6 smart management framework. Specifically, concepts such as certain AI enablers to support a data-driven architecture have been incorporated, as well as concepts regarding network modularisation, virtualisation and cloud transformation.
- Collaboration regarding the addressing of the Quantifiable Targets (QTs) assigned to this WP6. Beyond offline interactions, a specific workshop was organised regarding this the first week of December 2024. The information in Section 3.6.2 is in part a result from those interactions.

WP2:

- Consideration of the topics in the interim overall 6G system design presented in Deliverable 2.3 (released in M18, at the same time as WP6 D6.3 [HEX224-D23]), as well as the updated E2E system evaluation results from the interim overall 6G system, presented in Deliverable 2.4 [HEX224-D24].
- Sharing of information in the Workshop Session 6 (WS6) during the Hexa-X-II plenary meeting held in Den Haag (24-25 Sept. 2024). This WS6 was organised specifically targeting the M&O related topics, with presenters and specific sessions from both: WP6 and WP2. From WP6, the progress on the management framework (which was already quite close to the one presented in this deliverable) and its enablers was described. The alignment between this management framework and the E2E System Blueprint design being carried out in WP2 was also a topic of discussion.
- Also, certain enablers in the management framework have been designed to support the intent-based management approach being addressed in WP2 (specifically the Secure AI/ML-based control for Intent-based Management system and the Trust Management functionality).

- Contributions to include the multi-stakeholder concept in the context of this WP2, also towards the E2E System Blueprint design.
- Contribution to the KERs refinement process carried in WP2.

Regarding the alignment with the WP2 E2E System Blueprint, it is considered that the Smart Network Management Framework would be directly mapped in the so-called "Management & Orchestration" pervasive block, highlighted in red in Figure 3-2.

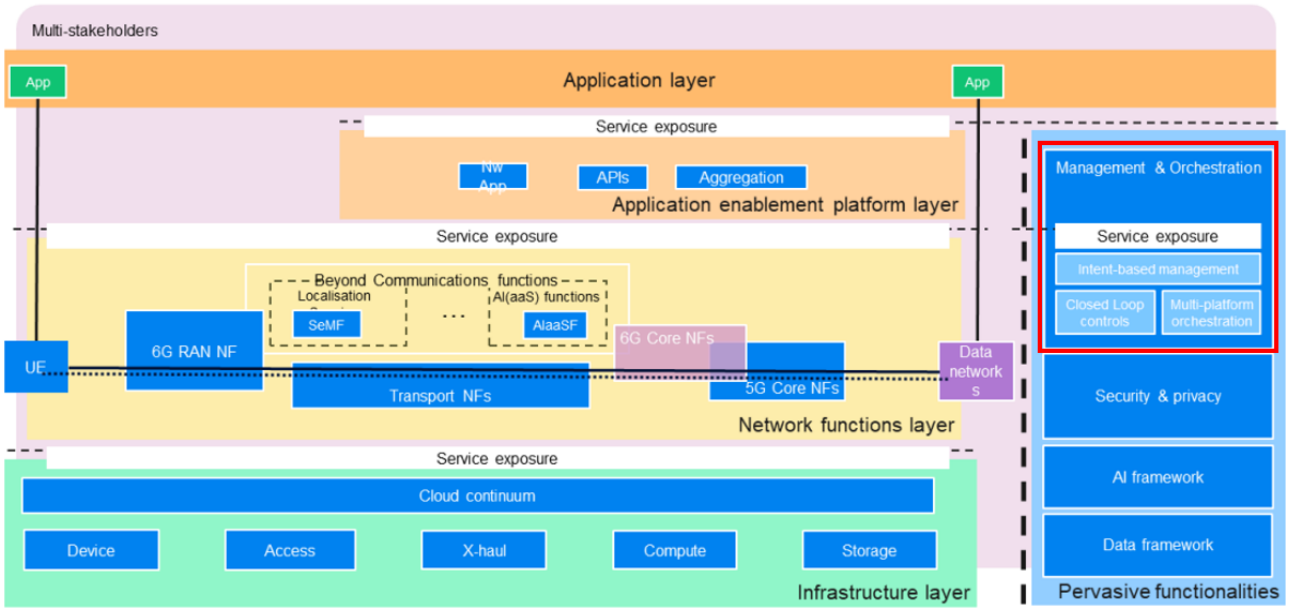

Figure 3-2: Alignment of the Smart Management Framework with the E2E System Blueprint in [HEX224-D24].

In this context, those overall functionalities and solutions in the Smart Management Network targeting the whole network continuum scope (components in the dark blue frame in Figure 3-2) would be acting as multi-stakeholder functions, i.e., those represented in the E2E System Blueprint by the overall pink rectangle that seems to be behind the whole picture. Anyway, further discussions on this mapping are scheduled to continue in the context of the WP2 once this Deliverable D6.5 is released and are planned to be reported in the upcoming WP2 Deliverable D2.5 (Final overall 6G system design).

3.2 Dissemination and standardisation activities

The following section provides an overview of WP6 dissemination and standardisation efforts, highlighting academic publications, open-source contributions, OpenAPIs, and related activities.

Table 3-1: Publications for WP6 Hexa-X-II during RP2

Title	Type of Publication	Status
Applying Digital Twins to Optical Networks with Cloud-native SDN Controllers	Article in Journal	Published
IntentLLM: An AI Chatbot to Create and Explain Slice Intents in TeraFlowSDN	Publication in conference proceeding / workshop	Published
Utilizing Causal Learning for Cognitive Management of 6G Networks	Publication in conference proceeding / workshop	Published
Exploiting Queue Information for Scalable Delay-Constrained Routing in Deterministic Networks	Article in Journal	Published
An East-Westbound Control Architecture for Multi-Segment Deterministic Networking	Publication in conference proceeding / workshop	Published
Secure AI/ML-based control in Intent-based Management System	Publication in conference proceeding / workshop	Published

Enabling Traffic Forecasting with Cloud-native SDN Controller in Transport Networks	Article in Journal	Published
Network Resource Allocation for Gaming Using MEC API and TeraFlowSDN	Publication in conference proceeding / workshop	Published
A Cloud-Native Approach for Orchestrating 6G-Enabled Services at the Computing Continuum	Publication in conference proceeding / workshop	Published
Towards an AI/ML-driven SMO Framework in O-RAN: Scenarios, Solutions, and Challenges	Publication in conference proceeding / workshop	Published
Attention to Virtualisation: Making Network Digital Twins aware of Network Slicing	Article in Journal	Submitted
Trust-based Intent Management for 6G: A Level of Trust Assessment Function	Article in Journal	Submitted
AI-Driven Orchestration of 6G-Enabled Services Across the Computing Continuum	Publication in conference proceeding / workshop	Submitted
Trust-based Intent Management for 6G: A Level of Trust Assessment Function	Article in Journal	Submitted
Performance model for managing Cloud-native Network Function deployments in closed-loops	Publication in conference proceeding / workshop	Submitted
A Privacy Protection Framework for Data Analytics in Network Management and Orchestration	Publication in conference proceeding / workshop	Submitted

Table 3-2: Open-source contributions for WP6 Hexa-X-II during RP2

Title	**SW Origin**	**Licence**	**Available in**	**PoC / Demo**	**Related Component / Use Case**
ETSI TeraFlowSDN	Existing SW - Enhanced in project	Apache 2.0	TeraFlowSDN Website	PoC B.1	Network Programmability
Digital Ledger Technology (DLT) for service federation	New SW - Created in project	Apache 2.0	GitLab	PoC B.1	Federated orchestration system
Infrastructure Layer Emulator (ILE)	New SW - Created in project	Apache 2.0	GitLab	PoC B.1	Decentralised Orchestration System
HX MLOps	New SW - Created in project	Apache 2.0	GitLab	Testbed	Sustainable MLOps
Routing protocols for deterministic networks	Existing SW - Enhanced in project	Apache 2.0	GitHub	Testbed	Network Programmability
6G latency sensitive service for Smart Manufacturing	New SW - Created in project	Apache 2.0	GitLab	PoC B.1	Overall M&O System enablers.

Table 3-3: OpenAPI for WP6 Hexa-X-II during RP2

Title	**SW Origin**	**Licence**	**Available in**	**PoC / Demo**	**Related Component / Use Case**
DLT Service Federation Open API	New - Created in project	CC Attr. 4.0 International	Zenodo	PoC B.1	Federated Orchestration
Sustainable MLOps models sharing API	New - Created in project	Apache 2.0	Zenodo	Testbed	Sustainable MLOps
TMF640 Service Activation Management API	Existing - Enhanced in project	Apache 2.0	GitHub Zenodo	Testbed	Network programmability
TMF644 Resource Function Activation Management API	Existing - Enhanced in project	Apache 2.0	GitHub Zenodo	Testbed	Network programmability
Conflict Detection for CLs Automation and Management API	New - Created in project	CC Attr. 4.0 International	Zenodo	Testbed	Real-time zero-touch control loops automation and coordination functionality

Monitoring jobs configuration API	Existing - Enhanced in project	Apache 2.0	Zenodo	PoC B.1	Monitoring and Telemetry
Query historical monitoring data API	Existing	MIT	InfluxDB site	PoC B.1	Monitoring and Telemetry
Multi-cluster extreme-edge resource orchestration API	New - Created in project	Apache 2.0	Zenodo	PoC B.1	Multi-agent system for multi-cluster orchestration
Closed Loop Governance – Catalogue API	New - Created in project	Apache 2.0	Zenodo	PoC B.1	Real-time zero-touch control loops automation and coordination functionality
Closed Loop Governance – Lifecycle Management API	New - Created in project	Apache 2.0	Zenodo	PoC B.1	Real-time zero-touch control loops automation and coordination functionality
Integration Fabric API	New– Created in project	CC Attr. 4.0 International	Zenodo	Testbed	Management Capabilities Exposure
Level of Trust Assessment Function API	New – Created in project	GPL 3.0	Zenodo	Testbed	Trust Management System
Trust Evaluation Function API	New – Created in project	GPL 3.0	Zenodo	PoC A	Trust Management System

Table 3-4: Dissemination activities for WP6 Hexa-X-II during RP2

Title	**Type of activity**	**Target audience**	**Objective**	**Date**
Participation in the working group “Trustworthiness of the 6G-Platform Program”.	Meeting	Research community, Industry, EU institutions, Business partners	Common understanding and comparison of Trustworthiness for 6G with other projects around the world like the 6G Platform Program Germany.	20/09/2024
Presentation at EuCNC Special Session: Jazz Networks: A proposal for deploying network services in the 6G cloud continuum.	Conference	Research community, Industry, business partners, Innovators, EU Institutions	Sharing ideas and proposal regarding the continuum orchestration concept towards 6G.	04/06/2024

Table 3-5: Communication activities for WP6 Hexa-X-II during RP2

Description	**Audience**	**Channel**	**Date**
News item on the Hexa-X-II website: Hexa-X-II Deliverable D6.3 focuses on the initial design of 6G smart network management framework	Research community	Website	01/07/2024
D6.3 – Initial Design of 6G Smart Network Management Framework added to the Hexa-X-II website	Research community	Website	01/07/2024
LinkedIn post on D6.3	Public	Social media	01/07/2024
FNS & Hexa-X-II joint workshop. Presentation on the Hexa-X-II Architecture Aspects: Foundations of 6G smart management and orchestration design.	Research community	Event	23/09/2024

Table 3-6: Standardisation activities related to 3GPP

Activity	Type of Activity	Standardisation Group	Specification	Date	Status
S5-232907: "Rel-18 CR 28.541 Fix vague issues in EP_Transport with Federated network Modelling"	Revision of an existing standard	3GPP SA5	TS 28.541	03/03/2023	Approved
S5-232948: "pCR TR 28.836 Add solutions for expresing service and slice profile requirements as intent expectations"	Elaboration of a new standard	3GPP SA5	TR 28.836	03/03/2023	Approved
S5-233058: "Proposed way forward for NSRULE isolation topic"	Revision of an existing standard	3GPP SA5		03/11/2023	Approved
S5-233092: "DP on relationship between NEST, URSP and ServiceProfile"	Revision of an existing standard	3GPP SA5	TS 23.501, TS 234,526, TS 28.541, NG.116	03/03/2023	Approved
S5-233896: "Add stage 3 for data type AvailabilityStatus"	Revision of an existing standard	3GPP SA5	TS 28.541	26/05/2023	Approved
S5-234698: "Discussion paper on isolation and sharing"	Revision of an existing standard	3GPP SA5	TS 28.541, TS 28.531	26/05/2023	Rejected
S5-234586: "Add NetworkSliceController and NetworkSliceSubnetController IOCs to support asynchronous LCM operations"	Revision of an existing standard	3GPP SA5	TS 28.541	26/05/2023	Approved
S5-234587: "Update Procedure of Network Slice Instance Allocation to support asynchronous operations"	Revision of an existing standard	3GPP SA5	TS 28.531	26/05/2023	Approved
S5-234588: "Update Procedure of Network Slice Instance deallocation to support asynchronous operations"	Revision of an existing standard	3GPP SA5	TS 28.541	26/05/2023	Approved
S5-234589: "Update Procedure of Network Slice Instance Modification to support asynchronous operations"	Revision of an existing standard	3GPP SA5	TS 28.531	26/05/2023	Approved
S5-234590: "Update Procedure of Network Slice Subnet Instance Allocation to support asynchronous operations"	Revision of an existing standard	3GPP SA5	TS 28.531	26/05/2023	Approved
S5-234591: "Update Procedure of ne"twork slice subnet instance deallocation to support asynchronous operations	Revision of an existing standard	3GPP SA5	TS 28.531	26/05/2023	Approved
S5-234592: "Update Procedure of Network Slice Subnet Instance Modification to support asynchronous operations"	Revision of an existing standard	3GPP SA5	TS 28.531	26/05/2023	Approved

S5-234716: "InputToDraftCR Rel-18 28.533 on Access control for management service"	Elaboration of a new standard	3GPP SA5	TS 28.533	26/05/2023	Approved
S5-234742: "Rel18 CR TS 28.541 Improve EP_Transport model to clarify connection point info"	Revision of an existing standard	3GPP SA5	TS 28.541	26/05/2023	Approved
S5-236015: "Rel-17 CR TS 28.312 Add missing stage 3"	Revision of an existing standard	3GPP SA5	TS 28.312	25/08/2023	Approved
S5-236016: "Rel-18 CR TS 28.312 Add missing stage 3"	Revision of an existing standard	3GPP SA5	TS 28.312	25/08/2023	Approved
S5-235664: "Discussion paper on GST version and Release"	Revision of an existing standard	3GPP SA5	TS 28.541	25/08/2023	Approved
S5-236048: "Rel-18 CR TS 28.554 Correct reference and fix void section"	Revision of an existing standard	3GPP SA5	TS 28.554	25/08/2023	Approved
S5-236049: "Rel-17 CR TS 28.554 Correct reference and fix void section"	Revision of an existing standard	3GPP SA5	TS 28.554	25/08/2023	Approved
S5-236050: "Rel-16 CR TS 28.554 Correct reference and fix void section"	Revision of an existing standard	3GPP SA5	TS 28.554	25/08/2023	Approved
S5-235170: "Rel-18 CR TS 28.541 Add NRM for network slice isolation"	Revision of an existing standard	3GPP SA5	TS 28.541	25/08/2023	Rejected
S5-235344: "pCR TR 28.836 Remove figure of how ServiceProfile can be represented by intent expectation components"	Elaboration of a new standard	3GPP SA5	TR 28.836	25/08/2023	Approved
S5-235946: "InputToDraftCR Rel-18 28.533 on Access control related to ConditionForMOIs"	Elaboration of a new standard	3GPP SA5	TS 28.533	25/08/2023	Rejected
S5-235947: "InputToDraftCR Rel-18 28.533 on Access control related to PermissionsForMnSs"	Elaboration of a new standard	3GPP SA5	TS 28.533		Approved
S5-237203: "DP on Service Management in SA5"	Elaboration of a new standard	3GPP SA5		13/10/2023	Approved
S5-237047: "Rel-18 CR TS 28.533 Add example of RAN domain management capabilities mapped with ZSM"	Revision of an existing standard	3GPP SA5	TS 28.533	13/10/2023	Approved
S5-237082: "Rel-18 CR TS 28.541 Add NRM for network slice isolation"	Revision of an existing standard	3GPP SA5	TS 28.541	13/10/2023	Rejected
S5-237174: "pCR 28.836 Enhance benefit description in 4.6"	Elaboration of a new standard	3GPP SA5	TR 28.836	13/10/2023	Approved

Table 3-7: Other related standardisation activities

Activity	**Description**	**Standardisation Group**
Intent-driven Closed Loops Introduction	ETSI ZSM intent driven closed-loop introduction	ETSI ZSM

Intent-driven closed loop and additional services	Governing intent-driven closed loop and additional services	ETSI ZSM
ETSI ZSM: Additional Services and Capabilities	This contribution is related specifies ZSM management services related to intent management	ETSI ZSM
Intent-driven Closed Loop Governance Service	Specifies additions to ZSM management service related to Closed Loop Governance	ETSI ZSM
Network resource allocation for Gaming using MEC BandWidth Management service and TeraFlowSDN	ETSI MEC Proof-of-Concept, based on research in T6.3	ETSI ZSM
Deterministic Networking (DetNet) Controller Plane Framework (draft-ietf-detnet-controller-plane-framework-05)	It provides a framework overview for the Deterministic Networking (DetNet) controller plane.	IETF DetNet WG
MIPv6 RAW mobility (draft-bernardos-detnet-raw-mobility-00)	This document discusses and specifies RAW/DetNet control plane solutions to cope with mobility, by proactively preparing the network for the change of point of attachment of a connected mobile node. It also defines Mobile IPv6 extensions implementing these control plane solutions.	IETF DetNet WG
Mobility challenges in virtualisation environments (draft-bernardos-dmm-mobility-virtualisation-02)	This document aims at presenting the new mobility-related scenarios (due to virtualisation) and the potential gaps in terms of IETF protocols	IETF DMM WG
An Evolution of Cooperating Layered Architecture for SDN (CLAS) for Compute and Data Awareness (draft-contreras-coinrg-clas-evolution-02)	Proposes an extension to the Cooperating Layered Architecture for Software-Defined Networking (SDN) by including compute resources and data analysis processing capabilities	IRTF COINRG

3.3 Contribution to the project Key Exploitable Results

The work performed during the reporting period in WP6 is considered to impact on the project Key Exploitable Results (KERs) defined in Table 3-8 below:

Table 3-8: WP6 KERs.

KER	**Rationale**
KER 2.1 – E2E system blueprint of the sustainable, inclusive, and trustworthy 6G platform.	The work performed in this WP6 is actually intended to support the E2E system blueprint design being addressed in WP2. Besides, the topics of sustainability and trustworthiness are specifically addressed by different enables in the Smart Management Framework.

<table>
<tr><td>KER 2.3 – Intent-based digital service management.</td><td>Though Intent Based Management (IbM) is a work topic specifically assigned to WP2 according to the Hexa-X-II work programme, the WP6 Smart Management Framework also considers this topic in certain technical enablers, specifically the following:
- In the “Secure AI/ML-based control system for Intent-based Management”.
- In the “Trust Management System”, which considers an E2E intent-based trust management solution to assess and ensure the trustworthiness of network services or resource provisioning.
- In the “Decentralised Orchestration System”, which also considers IbM as a way to define the deployment descriptors for the service components to be deployed.</td></tr>
<tr><td>KER 2.4 - Integrated Network Digital Twins for E2E security, privacy, and resilience assessment.</td><td>The Network Digital Twins technology is specifically addressed from the “Network Digital Twins Creation Mechanisms” system in the smart management framework.</td></tr>
<tr><td>KER 3.1 – Integration of AI in the 6G data-driven architecture.</td><td>AI/ML techniques are extensively applied in the Smart Management Framework. All those components in the framework highlighted with the sparkle icon (✨), which are most of them, are AI/ML based or related with AI/ML functions.</td></tr>
<tr><td>KER 3.2 – Easily deployable modular and scalable architecture</td><td>The Overall M&O Solutions in the management framework are considered well aligned with this KER.</td></tr>
<tr><td>KER 3.3 – Network of networks</td><td>All the enablers targeting the whole network continuum scope in the management framework are specifically aligned with this topic.</td></tr>
<tr><td>KER 3.5 – Cloud transformation</td><td>The M&O technical enablers in the management framework are intended to adapt the cloud for the 6G requirements, as requested for this KER.</td></tr>
<tr><td>KER 6.1 – Programmable flexible network configuration of transport networks</td><td rowspan="8">All these KERs are associated to the following specific components in the Smart Management Framework:
- The Network Programmability system, targeting KER 6.1.
- The Monitoring and Telemetry functionality, targeting KER 6.2.
- The Management Capabilities Exposure functionality, targeting KER 6.3.
- The Energy Efficient Service Function Allocation algorithms, targeting KER 6.7.
- The Trust Management functionality, the User-centric Service Provisioning system, and the 3rd Party Resource Control Separation system, all of them contributing to KER 6.9.
- The Network Digital Twins Creation Mechanisms system, targeting KER 6.10.
- The Rear-time zero-touch control loops automation & coordination functionality, targeting both KER 6.11 and KER 6.12.</td></tr>
<tr><td>KER 6.2 – Programmable network monitoring and telemetry.</td></tr>
<tr><td>KER 6.3 – Integration fabric.</td></tr>
<tr><td>KER 6.7 – Sustainable resource allocation for network management</td></tr>
<tr><td>KER 6.9 – Methods for trustworthiness of AI/ML in network management</td></tr>
<tr><td>KER 6.10 – Network Digital Twins for autonomous network management</td></tr>
<tr><td>KER 6.11 – Closed loop governance</td></tr>
<tr><td>KER 6.12 – Multiple Closed loops coordination</td></tr>
<tr><td>KER 6.4 – Resource controllability separation for multi-tenancy support</td><td>The 3rd Party Resource Control Separation system in the framework specifically targets this KER.</td></tr>
<tr><td>KER 6.5 – Multi-cluster resource management mechanisms</td><td>The Overall M&O Solutions in the management framework contribute to these two KERs.</td></tr>
</table>

KER 6.6 – Intelligent orchestration mechanisms for the computing continuum	
KER 6.8 – Methods for network management ML pipeline sustainability	The Sustainable MLOps system from the management framework contributes to this KER.

3.4 Alignment with the project use cases

The six use case families defined in WP1 (with their representative use case in brackets) are the following [HEX223-D12]:

1. Immersive experience (seamless immersive).
2. Collaborative robots (cooperating mobile robots).
3. Physical awareness (network-assisted mobility).
4. Digital twins (real time digital twins).
5. Fully connected world (ubiquitous network).
6. Trusted environments (human-centric services).

The *Immersive experience* use case family focuses on XR technology to create consistent immersive environments for telepresence, collaboration, education, gaming, and content creation by synchronising 3D visuals, spatial audio, and haptics. The *Collaborative robots* use case family involves intelligent, mobile robots that sense, perform tasks, and cooperate with humans or other robots in domains like house, healthcare, and manufacturing. The *Physical awareness* use case family combines sensing, positioning, and communication for physical scene analysis, tracking, trajectory prediction, and collision avoidance in scenarios with AGVs, drones, cars, and pedestrians. The *Digital twins* use case family focuses on the creation of interactive digital equivalents of the real world for control, maintenance, and management in domains like manufacturing, infrastructure, and communication networks. The *Fully connected word* use case family demands ubiquitous network access through terrestrial and non-terrestrial networks for services like crisis management, digital health, and autonomous supply chains. Finally, the *Trusted environments* use case family focuses on human-centric, privacy-protected services in local environments like hospitals or schools, using sensing and AI for spatial awareness and context-driven interventions.

The smart network management framework described in Section 2 is designed to support all the project use cases, as network and service M&O serves as a pervasive functionality essential for the practical realisation and deployment of any use case on the network.

Particularly, the framework targets the immersive experience use case family by providing seamless service continuity and ensuring privacy protection. It targets the collaborative robots use case family by contributing to enhanced automation and trustworthy interactions. The physical awareness use case family benefits from the smart network management framework by offloading tasks/workloads and ensuring privacy and safety. The digital twins use case family is targeted by the framework with the specific network digital twin creation mechanism, the privacy and trustworthiness guarantying. The fully connected world use case family is targeted by the framework with the network service assurance functionality, privacy protection, exposure capabilities, and more. Finally, the trusted environments use case family is addressed by the framework with the trust management functionality, privacy protection and secure control.

Section 2.2 showcases specific implementation examples targeting some of the use case families defined in WP1. Specifically, the collaborative robots use case family was studied in the *Functionality allocation in a cobot-powered warehouse inventory management*, the *AI-enabled RT zero-touch control loop analysis function* and the *Services orchestration over resources in the network continuum* implementations, described in Sections 2.2.2.3, 2.2.1.5 and 2.2.2.2, respectively. The fully connected world use case family is investigated in the *ML-based recommendation for energy management in service orchestration* implementation presented in Section 2.2.2.4. The trusted environments use case family is observed in the *Management Capabilities Exposure for Network Service Automation* implementation which can be found in Section 2.2.2.1.

3.5 Impacted KVIs

The Key Value Indicator (KVI) concept is a crucial component in the Hexa-X-II project targeting to address societal, environmental, and economic challenges while ensuring technological innovation. KVIs rely on the more general "Key Value" (KV) concept, which represents the impact of use cases and technology on the high-level human and planetary goals defined from the United Nations (UN)[6]. Based on this, a KVI represents a high-level qualitative or quantitative metric designed to measure the performance and alignment of technology development with broader societal values and project goals. Unlike traditional KPIs, which focus on technical parameters like latency, throughput, and energy efficiency, KVIs emphasise the value-driven and impact-oriented aspects of technology, such as sustainability, inclusiveness, and trustworthiness. KVIs serve to guide research and development toward achieving overarching societal and strategic objectives. They help ensure that the innovations in 6G are not only technically advanced, but also in line with the European Union goals for sustainability, digital inclusion, and global competitiveness. In Hexa-X-II the main work regarding this KVIs concept is driven from WP1 (Value, requirements and ecosystem). In coordination with this WP, in WP6 the set of KVIs that are considered relevant regarding network management have been identified as shown in the following set of Tables (from Table 3-9 to

Table 3-12) targeting the four different scopes of the Smart Network Management Framework, and on which:

- The left-hand column refers to the different components of the smart management framework, as defined in Section 2.1.
- The "Values as Goal" column represents the Human and Planetary Goals (Values as Goals/Criteria) identified as relevant values in WP1.
- The "Values as Outcome" column contains a brief text explaining the rationale for each of these goals.
- Finally, the "KVIs" column contains a list of possible Key Value Indicators for those values[7].

Table 3-9: KVIs regarding the Overall M&O Solutions

Component	Values as Goal	Values as Outcome	KVIs
Multi-agent system for multi-cluster orchestration	Ensure ICT Trustworthiness	Trust Management: Development of trust management schemes among the agents.	Adoption of agent-based mechanisms in orchestration solutions
	Foster Resilience	Increase automation, including self-healing functionalities.	High availability
	Build Long-term Sustainable Economic Growth	Novel business models based on the provision of optimal orchestration mechanisms.	Economic Development based on the emergence of novel orchestration solutions for the network continuum, increasing automation and reducing OPEX
Decentralised Orchestration system	Foster Resilience	Minimisation of network service downtimes even when they are deployed on volatile extreme-edge infrastructure resources.	Network services resiliency
	Minimise Waste	The incorporation of extreme-edge infrastructure resources would make possible to use	HW re-utilisation and sharing

[6] https://sdgs.un.org/goals.

[7] According to the project work plan, this work regarding the KVIs identification is scheduled to continue in the context of WP1 once this Deliverable D6.5 is released.

		certain devices when they are not being used for their primary function, e.g., while they are left in stand-by or, even operating, may have extra computational or storage resources still available. This may also help to reduce the size of centralised datacentres, so reducing CO_2 emissions.	
	Increase Digital Inclusion	The incorporation to the network of the extreme-edge devices placed in such areas could help to provide network services there. Those "beyond the edge" devices would be managed as an extension of the regular network infrastructure within the operator's own domain.	Number of extreme-edge resources in rural or low coverage areas added to the network
	Build Long-term Sustainable Economic Growth	New business models based on the integration of the extreme-edge infrastructure domain could be developed.	Common indicators to measure the economic growth in an economic sector such as the Gross Value Added (GVA), the Employment Growth, the Sector-Specific Output (e.g., focusing on the number of services delivered), or the contribution to the GDP
	Environmental Sustainability	Extreme-edge nodes already consuming energy but not hosting services could execute network service components, eliminating the need of connecting new devices, which would consume additional energy. Utilizing these extreme-edge resources could also reduce energy spent on data transmission by processing workloads locally, avoiding data transfer to central datacentres.	Common indicators to measure the electric energy consumption could be used, such as the Total Energy Consumption (kWh), Load Profile (kWh over time intervals), Energy Intensity (amount of electricity used per unit of output or activity), Carbon Emissions per kWh, etc.

Table 3-10: KVIs regarding the Overall Functionalities

Component	**Values as Goal**	**Values as Outcome**	**KVIs**
Monitoring and Telemetry	Foster Resilience	Continuous monitoring of infrastructure for quick reaction and adaptation against failures.	Service Resiliency
	Build Long-term Sustainable Economic Growth	Novel business models based on the provision of resilient infrastructure. Minimisation of Mean Time To Detect.	New business models

Real-time Zero-touch CLs automation and coordination	Foster Resilience	Management of CLs operating at the service layer for detection or prediction of possible failures in computing infrastructure, network connectivity or service applications with automated reactive or proactive mechanisms for end-to-end service recovery (e.g., via service migration, scaling, re-configuration).	End-to-end service resiliency
		Management of CLs operating at the network layer for detection or prediction of possible failures in computing or transport infrastructure and network functions, with automated reactive or proactive mechanisms for network resiliency (e.g., via protection or re-routing strategies at the transport domain, automatic selection of alternative RAN technology, self-healing of network services).	Resiliency of mobile network connectivity
	Minimise waste	CL functions are following a cloud native approach, and they can be deployed on-demand, where and when needed, on general purpose hardware. Their upgrade is performed automatically in software.	HW re-utilisation and sharing
	Increase Digital Inclusion	CLs can be specialised to automatically select and move to NTN connectivity (integrated in the mobile network) to reach remote areas, possibly adjusting the network configuration to the service requirements.	Extension of network coverage
	Ensure ICT Trustworthiness	The definition of common models and interfaces for CLs and CL functions (possibly mediated via MCE) allow interoperability among CLs from different vendors.	Increase collaboration among technology providers
Management Capabilities Exposure	Ensure ICT Trustworthiness	The MCE fosters trust by securely managing APIs and ensuring transparent interactions. This visibility builds user confidence in ICT platforms.	Growth in user confidence in digital platforms due to visible trust management systems

		MCE enables secure and reliable communication across stakeholders, creating an environment where collaboration is seamless and trust driven.	Increased cross-sector collaboration facilitated by trusted ICT infrastructure
	Ensure Transparency	The MCE provides event-driven communication and structured API interactions, ensuring fairness and openness across networks, fostering trust among stakeholders.	Increased stakeholder satisfaction with the fairness and openness of inter-network operations
		By providing clear, observable integration mechanisms, MCE helps network providers demonstrate their commitment to transparency, improving public perception.	Higher public approval ratings for network providers due to transparent communication practices
		MCE's compliance with standards like ETSI ZSM promotes a unified, transparent approach to global network governance, facilitating cooperation across stakeholders.	Enhanced global cooperation on network governance due to standardised and transparent frameworks
	Foster Resilience	Through secure and efficient management capabilities, MCE ensures system stability and accelerates recovery during network failures, supporting business continuity.	Reduction in downtime and faster recovery in ICT services during disruptions
	Increase Digital Inclusion	The MCE's scalable, event-driven architecture allows for the inclusion of diverse stakeholders, enabling equitable access to services, particularly for marginalised communities.	Higher adoption of digital services among underserved populations
	Curb Climate Change	MCE promotes efficient management and orchestration of resources, reducing redundant usage and driving the adoption of sustainable ICT practices.	Growth in adoption of energy-efficient ICT systems
SLA-driven Federated Orchestration	Ensure Transparency	The use of digital ledger technologies ensures that the transactions underpinning the establishment and fulfilment of contractual obligations are visible to all interested parties.	Guarantees an open, verifiable method of contracting providers
	Ensure ICT Trustworthiness	The tamper-proof nature of blockchains provides a	Provides a tamper-proof platform in which

		guarantee that the actions taken by the functionality can all be audited ensuring that the smart contract decisions can be properly policed.	transactions can be monitored and audited
Trust Management	Ensure transparency	LoTAF promotes the use of a transparency service to share trust assessments between all involved parties by using signed statements and cryptographic verification.	Level of credibility on trust assessment
	Ensure ICT Trustworthiness	LoTAF introduces a cutting-edge trust evaluation mechanism to consider trust as a new feature that can support the management of intent-based network solutions.	Enhanced trustworthy business establishments in multi-stakeholder and multi-domain scenarios.
		Trust evaluation functions provide trust indexes of compute nodes with advanced monitoring mechanisms and trust quantification for trustworthy workload placement used by cloud orchestration engines.	Higher preference of entities/compute nodes of higher trustworthiness characteristics (availability, reliability, security etc.) for executing computational tasks/workloads

Table 3-11: KVIs regarding the Specific Systems.

Component	**Values as Goal**	**Values as Outcome**	**KVIs**
3rd-party resource control separation	Ensure ICT Trustworthiness	Effective resource separation builds trust, facilitating broader cooperation.	Growth in collaborative initiatives between network providers and third-party services
		Separating resources minimises security risks and fosters trusted relationships.	Reduced incidents of misuse or security breaches involving third-party resource access
	Preserve Natural Resources	Clear boundaries and agreements ensure third parties operate sustainably, contributing to resource conservation.	Increase in resource-sharing agreements prioritizing energy-efficient infrastructure use
	Ensure Transparency	Transparent resource policies improve compliance and reduce disputes.	Increased number of third-party providers meeting transparent resource usage agreements
		Clear policies and transparent execution enhance stakeholder trust and engagement.	Stakeholder satisfaction rates regarding clarity in resource sharing policies
User-centric service provisioning	Foster Digital Inclusion	User-centric approaches reduce barriers for marginalised users by prioritizing accessibility.	Percentage increase in digitally underserved populations accessing services

		Personalised service provisioning encourages ecosystem growth focused on local needs.	Growth in regional/local content and applications developed for underserved users
	Ensure ICT Trustworthiness	User-centric systems empower users with data control, enhancing privacy.	Increased trust in digital services through end-user control over personal data usage
		Reduced unauthorised, unethical, or unintended use of personal data in service systems, ensuring compliance with privacy laws and user consent	Reduced instances of data misuse in service provision systems
Network Digital Twins Creation Mechanisms	Ensure ICT Trustworthiness	Increase reliability of networks due to the ability to predict impact before taking operational actions.	Accuracy of model output relative to its physical counterpart
Sustainable MLOps	Curb Climate Change	Optimizing ML operations reduces emissions, aligning with climate goals. Facilitates sustainable AI adoption.	Reduction in carbon emissions (CO_2 tons per model)
		Energy-efficient ML pipelines reduce the environmental footprint of AI systems.	Energy consumption during model training (kWh)
	Ensure ICT Trustworthiness	Secure MLOps environments build trust in AI systems and reduce operational risks.	Number of successful security audits
		Strengthens user confidence by actively addressing privacy and security concerns.	Frequency of vulnerabilities detected and resolved
	Minimise Waste	Regular optimisation updates ensure that software is running as efficiently as possible, reducing unnecessary resource consumption.	Frequency of software updates for optimisation and efficiency
		Promoting software reusability reduces duplication of effort and resources, contributing to more sustainable development practices.	Percentage of reusable software components (e.g., libraries, modules)
Network programmability	Build Long-term Sustainable Economic Growth	Novel business models based on the provision of new connectivity services.	New business models
	Ensure ICT Trustworthiness	Centralised control in SDNs can enable stronger and more dynamic security measures, fostering trust in digital systems. Besides, SDNs can improve network resilience and fault tolerance, ensuring	Increased trust in digital services

		consistent and dependable connectivity.	
	Environmental Sustainability	more efficient use and update of network resources, helping to reduce hardware dependencies and energy consumption. Also, automated updates and configurations can extend the lifespan of hardware, reducing energy waste.	Energy efficiency
Privacy protection for data analytics	Ensure Privacy in multi-vendor environment	Enhance privacy by proposing a framework that protect data used by analytics functions in multi-vendor environment**.**	Less disclosure of sensitive data in multi-vendor environment
Secure AI/ML-based control for Intent-based Management	Ensure security	Improving network resilience to cyber-attacks, making it more trustworthy.	Robustness of the decisions taken by AI-driven agents in the intent-based management system

Table 3-12: KVIs regarding the selected management algorithms.

Component	**Values as Goal**	**Values as Outcome**	**KVIs**
ML based configuration recommender for energy savings	Minimise Emissions and Disposals to Water, Air and Soil	Minimizing the impact of technology operation, using the minimal amount of energy needed to deliver services. This also decreases the emissions (in case of unclean generation) used to generate that energy.	Energy used for service delivery
Efficient network and service function allocation	Minimise Emissions and Disposals to Water, Air and Soil	Minimizing the impact of technology operation, using the minimal amount of energy needed to deliver services. This also decreases the emissions (in case of unclean generation) used to generate that energy.	Energy used for service delivery
Resource assignment for federated learning	Build Long-term Sustainable Economic Growth	Responsible consumption of resources, using the minimal amount needed to deliver services.	Resources used for service delivery
Multi-agent RL for adaptive scaling	Minimise Emissions and Disposals to Water, Air and Soil	Optimised usage of resources to accommodate the workload needs, avoiding the activation of multiple resources that may not be adequately utilised.	Energy efficiency
Explainability for RL-based control	Ensure ICT Trustworthiness	Improving system transparency and safety, making it accountable and more trustworthy for use.	Explainability of system actions

3.6 WP6 contribution to the Hexa-X-II objectives

The Hexa-X-II project has defined a number of objectives in its work programme, from which three were linked to WP6. They are the following:

- Objective 2 - Develop and describe the 6G platform on system level and evaluate it considering the requirements on 6G services.
- Objective 4 - Develop and describe solutions for an expanded scope of wireless networks, for creation and processing of data, considering the requirements on 6G services.
- Objective 5 - Develop and describe solutions for building the 6G platform considering the requirements of 6G services.

To address these overall objectives different WP6-specific objectives were also defined, namely:

- Targeting Objective 2:
  - **WPO 6.1**: Design and develop a programmable cloud-native micro-service-based Management and Orchestration (M&O) framework for the future 6G networks.

- Targeting Objective 4:
  - **WPO 6.4**: Design and implement robust and trustworthy AI/ML-based network control solutions with optimal energy efficiency and sustainability target.
  - **WPO 6.5**: Design and develop zero-touch M&O mechanisms for closed loop automation and continuous service assurance, guaranteeing compliance with relevant 6G KPIs while reducing OPEX.

- Targeting Objective 5:
  - **WPO 6.2**: Design and develop mechanisms that collectively define a 6G enabled trustworthy environment, with a user-centric integration fabric that ensures multi-tenancy support and SLA verifiability.
  - **WPO 6.3**: Develop synergetic orchestration mechanisms for managing the deployment of 6G services over heterogeneous resources across the IoT-to-edge-to-cloud continuum.

WPO 6.1 (the design and development of the M&O framework towards 6G) is considered the main outcome of the whole WP6 itself. It is considered fulfilled, with the design of the overall M&O framework provided in this D6.5, described in Section 2.1, and with the different developments described in Section 2.2.

WPO 6.2 (providing of a trustworthy environment with a user-centric integration fabric for multi-tenancy support) is also considered fulfilled, in what regards the integration of the Management Capabilities Exposure functionality as part of the M&O framework (described in Section 2.1.2.1), which in fact implements an Integration Fabric component enabling multi-tenancy support. Implementations based on this component have been also performed, as described in Section 2.2.2.1).

WPO 6.3 (regarding the development of synergetic orchestration mechanisms to deploy 6G services on the IoT-to-edge-to-cloud continuum), is also considered fulfilled in what regards the integration of two different M&O strategies (hierarchical and distributed) as part of the M&O framework (described in sections 0 and 0). An implementation of this feature is also described in Section 2.2.2.2.

WPO 6.4 (regarding the design and the implementation of robust and trustworthy AI/ML-based network control solutions targeting optimal energy efficiency and sustainability) is also considered fulfilled by providing different AI/ML systems and algorithms as part of the M&O framework (those described in Sections 2.1.3.7, 2.1.3.4, 2.1.4.1, 2.1.4.2, and 2.1.4.4). Implementations of some of these algorithms have been also performed (Section 2.2).

WPO 6.5 (regarding the design and development of zero-touch M&O mechanisms for closed loop automation) is also considered fulfilled by means of the real-time zero-touch control loops automation and coordination system part of the M&O framework, described in Section 2.1.2.2. An implementation of this system has been also performed. The implementation of the overall system was described in D6.3 [HEX223-D63] and it has been now extended to integrate with the Management Capability Exposure (see Section 2.2.2.1). Moreover, examples of closed loops implementations applied to different scenarios and objectives are reported in Sections 2.2.1.3, 2.2.1.4, 2.2.2.4, and 2.2.2.6, together with examples of closed loops coordination for conflict detection and mitigation described in Sections 2.2.1.1 and 2.2.1.2.

3.6.1 WP6 measurable results towards Objectives 2, 4, and 5

According to the Hexa-X-II workplan, measurable results linked to Objectives 2, 4, and 5 (i.e., those related to WP6), are:

Objective 2:

a) Published reports with 6G platform design descriptions. This is considered fulfilled based in this and the previous WP6 deliverables. Besides, several of the dissemination activities reported in Section 3.2 are also considered contributing to this objective.
b) Published reports with initial performance results from 6G system simulations. This is considered fulfilled with the previous Deliverable D6.3 (which anticipates some early evaluation results) and this Deliverable D6.5, where some additional results are provided, as well as the reporting of those Quantifiable Targets (QTs) assigned to this WP6. Besides, according to the Hexa-X-II work programme, more results will be provided in the context of WP2 in relation to the project PoCs in Deliverable D2.6 (Final end-to-end system evaluation results of the overall 6G system design), where additional concepts addressed in this WP6 are being integrated.
c) Three industry-leading proof-of-concept demonstrations covering selected innovative functionality of the 6G platform. This objective is considered fulfilled beyond the target objective. On one hand, early PoC demonstrations were presented at the 2024 EuCNC and 6G Summit conference, in the Hexa-X-II booth. Besides, more complete demonstrations are in execution while this document is being written (those described in Section 2.2) that are also planned to be presented at the 2025 EuCNC and 6G Summit conference, as part of the final project PoCs. Also, one of these proof-of-concepts was presented at the 2024 Brooklyn 6G Summit conference.

Objective 4:

a) Proof-of-concept demonstration of a data exposure service. The WP6 management framework exposes management capabilities. This has been addressed by including the Management Capabilities Exposure functionality in the framework. An example implementation of this functionality has been also performed, described in Section 2.2.2.1.
b) Published report describing 6G network compute-AI service concepts with simulation results. This has been fulfilled with the previous Deliverable D6.3 which provided early evaluation results, and this Deliverable D6.5 which provides additional simulation results.
c) Published report describing 6G network sensing service concepts with simulation results. This is considered out of scope for WP6, since it refers a very specific network service (a sensing service). However, report [HEX224-D33] generated in the context of WP3 addresses this topic.

Objective 5:

a) Published report with a description of 6G implementation aspects with simulation results on resource-efficiency performance. This is considered fulfilled with the previous report [HEX224-D63], and this D6.5, specifically in what regards the following implementations:
   - Sustainable MLOps (Section 2.2.1.5).
   - Services orchestration over resources in the network continuum (Section 2.2.2.2).
   - Functionality allocation in a cobot-powered warehouse inventory management (Section 2.2.2.3).
   - ML-based recommendation for energy management in service orchestration (Section 2.2.2.4).
   - Resource assignment for federated learning (Section 2.2.2.5).
b) Published report with a description of a 6G network management concept. The different management concepts associated to the WP6 smart management framework are described in this and the previous WP6 reports. Also, in other reports from other WPs, e.g., where some of these concept have been conceptually presented (e.g., in WP3 deliverables).
c) Proof of concept demonstration of 6G implementation. Although according to the Hexa-X-II project plan this is specifically assigned to WP2, this WP6 has also contributed to the demonstration activities. On one hand, by interacting with the WP2 to integrate certain concepts developed in this WP6 in the WP2 PoCs, which is to be reported in the WP2 deliverables D2.5 and D2.6. However, beyond that,

certain WP6 specific implementations have been done on in-house testbeds to test and validate some of the concepts that have been developed in this WP6. The implementations described in the previous Deliverable D6.3, and in Sections 2.2.1 and 2.2.2 in this Deliverable, illustrate the focus on performing practical implementations and PoCs in the context of this WP6. As mentioned above, early versions of these implementations were showcased at the 2024 EuCNC and 6G Summit conference, and more complete demonstrations are also planned to be presented at the 2025 EuCNC and 6G Summit conference.

d) Proof of concept demonstration of 6G network management. Specific management related PoCs were also presented at the 2024 EuCNC and 6G Summit conference, covering the following specific assets:
   - The Decentralised Orchestration system.
   - The Multi-agent system for multi-cluster orchestration system.
   - The SLA-driven Federated Orchestration.
   - The Real-time zero-touch control loops automation and coordination functionality.
   - The Trust Management functionality.
   - The Network Programmability system.
   - The Efficient network and service function allocation algorithm

3.6.2 Quantifiable targets towards the project objectives

Following the Hexa-X-II workplan, Objectives 4 and 5 also require verifying the following Quantifiable Targets (QT):

Objective 4:

- **QT 4.1**: (<1 m at 90th percentile) Radio/Communication based sensing precision in mid-band and (<10 cm at 90th percentile) Radio/Communication based sensing precision at 100 GHz to detect a moving human sized object at 10 m distance.
- **QT 4.2**: (>20%) improvement in performance in at least one of energy efficiency, latency, bit rate or area capacity through use of sensing, localisation, traffic data, or mobility patterns for AI-based optimisation in selected use cases.
- **QT 4.3**: Trustworthy communication and compute network services for distributed AI applications in large scales (applications with >1000 collaborating AI components).

Objective 5:

- **QT 5.1:** Reducing energy consumption per bit in networks by (>90%).
- **QT 5.2:** (>25%) reduction in OPEX by using zero-touch automation.

From these, WP6 has been responsible for verifying those QTs with a clear relationship with the smart management topic, namely QT 4.2, QT 4.3, and QT 5.2. The following subsections analyse how the WP6 specific QTs have been verified. The verification approach involved defining particular baselines, based on the project PoCs or certain lab experiments/simulations. Any claimed improvements are over the explicitly stated baselines and may not extrapolate over all the considered cases.

3.6.2.1 QT 4.2. Improvement in performance

This QT requires >20% improvement in performance in at least one of energy efficiency, latency, bit rate or area capacity through use of sensing, localisation, traffic data, or mobility patterns for AI-based optimisation in selected use cases.

Many work items carried out in this WP6 contribute to this target but their contributions could not be quantified because it was not possible to establish reasonable baselines. Here we list those to which the contribution can be expressed in reference to a stated baseline, namely:

a) The resource allocation mechanism described in section 2.1.4.2 provides metaheuristics based on a genetic algorithm and ant colony optimisation for optimizing the placement of computation workloads and tasks with the goal of minimizing energy consumption, end-to-end latency and trustworthiness. The algorithm uses traffic data, localisation of users and compute nodes, as well as user mobility patterns as input to provide resource allocation decisions. In a specific scenario with 50 fixed compute

nodes, the proposed mechanism for computational workload placement resulted in a 9-70% decrease in power consumption depending on the number of workloads, compared to a round-robin placement algorithm which matches workload functional requirements to node features. In another scenario involving 30 AMRs and UAVs involving mobility, the allocation mechanism for physical task planning used an ant colony optimisation algorithm was compared with a nearest neighbour heuristic and showed a 27.9-35.9% energy consumption reduction and 25.1-60% less time to complete (i.e. computation latency) depending on the number of physical tasks.

b) Considering that RAN contributes for about 80-85% of the overall energy consumption, the ML-based configuration recommender system for base stations described in section 2.1.4.1 can be used to perform dynamic optimisation of the base stations configuration of cells' operational mode (active vs. sleeping mode). It needs user traffic data dynamics and Variational Autoencoders (a type of GenAI models), with the aim of minimizing power consumption while maintaining the QoS metrics for the users. In the referenced study, the proposed mechanism exhibited a 10-12% improvement in energy consumption of base stations (measured in Watts) compared to the static and/or manual network configuration methods that are not adaptive to changing conditions of the network. The proposed strategy, based on closed loop configurations differentiated on a per-cell basis, was able to reach the reported level of energy consumption decrease without any degradation in terms of performance (RRC, ERAB success rates, or call drop rates) or DL/UL throughput and maintaining stable traffic volumes and mobility success rates.

c) Finally, the algorithm presented in Section 2.1.4.4 details a multi-agent Reinforcement Learning approach to perform dynamic adaptive scaling of applications in a multi-provider setup, where the training of the model is centralised with a decentralised execution by the distributed agents. The proposed solution considers the individual service and system measurements (end-to-end latency, traffic data such as workload rate and throughput) and determines a decision policy that translates them into efficient scaling decisions for optimizing towards the intended service requirements. Applying the obtained scaling policy which uses traffic data dynamics resulted in a 10% decrease of end-to-end latency SLA violations compared to K8s HPA, while using 20% less resources.

Based on these assessments, it can be observed that the cited solutions within this deliverable employed AI-based mechanisms to provide improvements in performance in energy efficiency and latency by using traffic data. Considering the improvement values, which are between 10-70% depending on the scenario, the target can be considered fulfilled.

3.6.2.2 QT 4.3. Trustworthy communication and compute network services

This QT requires trustworthy communication and compute network services for distributed AI applications in large scales, with applications with >1000 collaborating AI components.

Albeit not in a quantifiable improvement over a baseline, the work described in 2.1.2.5 Trust Management contributes to the trustworthiness of communication services by defining a metric called Trust Index as a weighted sum of Availability, Reliability, Security, Multi-connectivity capabilities and Battery level, which is used to evaluate the trust of the infrastructure and network within M&O procedures. Furthermore, the work described in sections 2.1.3.6 Privacy Protection for data analytics in M&O and Explainability for RL-based Control both contribute to the trustworthiness of network management and, as a consequence communication, but their contributions could not be quantified.

Below is a list of work items whose contribution can be expressed in reference to a stated baseline, namely:

a) The solution described in section 2.1.3.7 Secure AI/ML-based control for Intent-based Management contributes to the trustworthiness of communication by making the AI/ML model used for decision making more robust. It provides a security mechanism to protect intent management systems against adversarial attacks by using adversarial training techniques. Under adversarial attacks, the trained models exhibited better performance compared to models without adversarial training. The Mean Squared Error (MSE) for the models trained with adversarial training was approximately 30% less than the MSE for the baseline models.

b) The solution in section 2.2.2.5 Resource assignment for federated learning facilitates in-network compute services which enabled a 37% decrease in communications overhead compared to [YLC+21], in a scenario involving 9 domains with one data source and one computing element per domain. The result was achieved by reusing existing communications when possible and optimised clustering, which allowed for up to 30% fewer hops per epoch between learning peers and 10% fewer epochs required to attain convergence. Additionally, for each node, the process does not compare its dataset against all other nodes in the architecture, but only with the dataset of its neighbours, which means that the complexity of the algorithm is determined by the number of nodes and the average number of neighbours for each node. Although tests were only performed in small scales, the improvements do allow large scale ups.

Based on these assessments, it can be observed that the cited solutions improved the trustworthiness of communication with a 30% increase in robustness of the model used in decision making, and the compute network services with a 37% decrease of communication overhead, 30% fewer hops per epoch between learning peers and 10% fewer epochs overall, allowing the solution to scale up. However, the number of collaborating components was not verified to be >1000, so the overall QT can be considered partially fulfilled.

3.6.2.3 QT 5.2. Reduction in OPEX by using zero-touch automation.

This QT requests to verify the possibility to achieve a (>25%) reduction in OPEX by using zero-touch automation techniques. The assessment of this QT has been addressed in two different scopes, namely:

a) The set of orchestration mechanisms developed to improve automation in decision-making in the deployment and management of network services over infrastructure deployed in the computing continuum. Part of the developed mechanisms follow a multi-agent orchestration approach, as detailed in the algorithm detailed in Section 2.1.4.4 and in the implementation in the Component PoC#B.1 (Section 2.2.2.2). In this regard, it has been verified that through the collaboration of agents to support the deployment and scaling of a network service (Independent Scaling - Migration -ISM- scenario) it is possible to achieve low percentages of SLA violations (~2%) while optimally using the available resources. Given that decision-making is provided by DQN agents, and that the efficiency in terms of resources usage is high (5% consumption of resources compared to 50% in the case of one agent), it can be claimed that OPEX reduction is higher than 10%, due to the reduction in the operational and energy costs related to the computational infrastructure. However, more detailed trials and evaluation results are envisaged to be produced for this assessment in the context of the WP2 activities. In related studies [ZFF24] that examine the performance of multi-agent RL techniques for automated scaling, it is shown that by applying the obtained scaling policy resulted in a 10% end-to-end latency and request rate improvement, while using 20% less resources.

b) The examples of closed loops as well as closed loops coordination processes, reported in sections 2.2.1.1, 2.2.1.2, 2.2.1.3, 2.2.1.5, 2.2.2.4 and 2.2.2.6, which demonstrate the feasibility to automate several operational actions in network infrastructures reducing the need of human intervention. This has been demonstrated for scenarios related to resource allocation, service provisioning, automated recovery, as well as reconfiguration of cells' operation mode from active to sleeping mode and vice versa, so targeting energy consumption reduction. The latest example is directly associated to OPEX reduction, with 10-12% gain in the energy required for the RAN segment. More in general, zero-touch network automation mechanisms, whose feasibility has been extensively proved through Hexa-X-II implementations, help in reducing OPEX in terms of decreasing labour costs, maintenance time and costs, potential human errors in routine operational actions as well as increasing resource allocation efficiency and accelerating service provisioning, configuration and other lifecycle management actions like autoscaling and migration. For a quantification of the actual reduction in OPEX due to the introduction of network automation techniques, we can refer to the literature where several studies from big industrial players confirm the trend. For example, [Adl22] reports an increasing efficiency up to 20% achieved by European Telco players through the introduction of zero-touch network automation, with repetitive network maintenance activities that can be automated in the range of 70%-80% and up to up to 80% effort reduction in engineering tasks. Still in this direction, a recent report from Capgemini [Cap24] analyses the benefits from autonomous networks stating that telcos have

realised a 20% improvement in operational efficiency and 18% network OpEx savings, on average, through autonomous networks initiatives undertaken in 2022-2023, while 71% of telcos have reduced energy consumption in the same period. Moreover, the analysis suggests that OpEx savings from autonomous networks would be $150 million–$300 million per organisation over the coming five-year period, with an estimated ROI of 1.7x–3.4x and a payback period of 2.9 to 1.5 years in conservative and optimistic scenarios, respectively.

c) The MCE brings an implementation of the ETSI ZSM Integration Fabric, based on an event driven architecture. Adoption of event-driven architecture with Apache Kafka for microservices coordination participates in the reduction of OPEX by enabling asynchronous, decoupled communication. This minimises system-wide failures and streamlines scalability without manual intervention. Integration with Apache Kafka automates real-time analytics (e.g., fraud detection), cutting labor costs by ~30% [DZONE20]. Concurrently, ETSI ZSM integration fabric standardises zero-touch automation across multi-vendor networks, reducing manual configuration efforts by 70% and energy costs by 10–12% in RAN segments [ZSM-002][ZSM-003][ZSM-009-1]. Together, Kafka and ZSM enable end-to-end automation, lowering OPEX through resilient resource allocation (5% resource usage vs. 50% in legacy systems) and proactive SLA adherence (~2% violation rates).

In summary, it can be concluded that the network automation mechanisms proposed in Hexa-X-II can provide a strong contribution to the OPEX reduction QT, with up to 20% of reduction fully demonstrated, even though not able to achieve the entire 25% reduction when not applied in combination with other enablers.

3.7 Alignment with the Advisory Group recommendations

Below, the recommendations received from the Hexa-X-II Advisory Group that were specifically addressed to WP6, or that are considered relevant from the WP6 perspective:

- **Rec.1**: Attention should be paid on sustainability values, and inside that, also on social sustainability like inclusion and trust.

  Answer:
  Based on the input in Section 3.5 (Impacted KVIs) it is considered that WP6 is in line with this recommendation. As it can be appreciated, sustainability values have been considered for all the enablers, and some of them specifically addressing the increase of the digital inclusion and to ensure ICT trustworthiness.

- **Rec. 2**: To show contrast and improvement in 6G compared to 5G. What cannot be done in 5G? What can be done better?

  Answer:
  In the scope of WP6, the following improvements can be observed:
  - M&O solutions for the continuum orchestration. This extends the M&O mechanisms beyond the operator own infrastructure and enables operators to deploy and operate network services across domains, as well as integrate with third-party infrastructure in addition to their own.
  - DevOps and related methodologies are being adopted in more processes and on larger scales but also improved. One improvement coming from the work in WP6 is Sustainable-MLOps, described in Section 2.1.3.4 which makes the operations for ML pipelines energy efficient.
  - The uptake of AI/ML for complexity management in the context of the continuum management. AI/ML algorithms are more prevalent and natively integrated in an increasing number of M&O processes, like intent-based networking, resource efficient deployments, and other innovative functionalities, like those tagged with the icon (✨) in the management framework figure.
  - Dynamic integration of new infrastructure resources in the network. The implementation of the network continuum concept makes it necessary to enable mechanisms to manage and keep record of the dynamic and diverse resources beyond each stakeholder own domain (which could be highly volatile or even error prone). The improvements in the overall M&O solutions in this WP6,

as well as in the Real-time zero-touch control loops automation and coordination functionality further advance the concept.
- Integration of Network Digital Twins in M&O. NDTs are accurate models that can improve M&O procedures by providing the ability to test and verify different actions without acting on the production infrastructure. Their integration in the M&O is essential for closing the loop and moving away from automation towards autonomy.

- **Rec. 3**: Even if the corresponding work is mainly done in WP2, it is important to highlight the security aspects more in the context of this WP6.

  Answer:
  The security aspect has been highlighted in the WP6 smart management framework figure (Figure 2-1) by including a specific M&O-related functionality specifically referring Security (numbered as 8 in the central legend of the figure). As it can be appreciated, this number has been assigned to different technical enablers in the framework: (i) the real-time zero-touch control loops automation and coordination functionality, which can be applied to security orchestration with the aims of applying automated mitigation, remediation and recovery actions in case of security threats or attacks, (ii) the 3rd-party resource control separation system, which provides segregated management spaces for stakeholders in multi-stakeholder 6G environments, ensuring secure resource control and privacy, (iii) the secure AI/ML-based control for intent-based management system, which supports intent-based management systems to enhance their security, (iv) the Trust Management functionality, which can be used to integrate trust evaluation as part of the M&O systems targeting to ensure secure and efficient resource allocation in multi-stakeholder environments, and (v) the user-centric service provisioning system, which enables flexible and more secure SLA definitions. Beyond this, the security aspect has been also highlighted by means of one of the implementations presented in this document, specifically in the one showcasing the usage of the MCE functionality described in Section 2.2.2.1, where RBAC and secure APIs are used to ensure that only authorised entities can perform critical operations such as new entities and communication channels onboarding procedures, service reconfiguration, fault recovery, and trust levels assessment.

- **Rec. 4**: Beyond the traditional way to automate network management and operations, it is considered quite important to figure out how to improve the capability to monetise the network, and how the network capabilities could be expanded beyond communication, since traditional communication is only a lowest level expectation from customers.

  Answer:
  Although the business aspects topic is out of scope in this technical deliverable, the reaching of those network domains beyond the operator's own network domains, i.e., the integration of the extreme-edge domain in the M&O processes proposed from this WP6, is also considered a way to implement new and profitable business models [Law24]. Besides, it is considered that this approach not only optimises revenues, but can also reduce operating costs. The interoperability enabled by architectures based on microservices and APIs can provide multiple tangible benefits, such as market expansion and cross-sector collaboration, the elimination or access to specific technological silos, the deployment of services in new geographical areas and markets, and the reduction of time-to-market, among others. Furthermore, the emergence of orchestration approaches to manage both compute and network resources enable service providers to increase the automation and decentralised intelligence in their service provision and, in parallel, achieve OPEX reduction.

- **Rec. 5**: In line with some considerations within the GSMA, is recommended to consider on how to provide Open APIs to expose functionalities to external players.

  Answer:
  It is considered that WP6 contributes to this in different ways: on one hand, the "edge convergence over federated resources for the computing continuum" implementation (Section 2.2.2.7) explicitly explores

the capabilities of the GSMA CAMARA EdgeCloud APIs in the management of the compute resources in the network continuum, and the possibility to extend them to be used with federated resources of external administrative domains. Besides, the WP6 smart management framework Management Capabilities Exposure (MCE) functionality, which falls outside the GSMA's defined scope (it relies on the ETSI ZSM Integration Fabric concept), offers a valuable additional approach to improving interoperability and service management in the telecom's operator domain. It provides event-driven interfaces at the service level, which is similar to the approach taken by the GSMA, which considers facilitating interaction between different systems and external entities (although from slightly different perspectives). Finally, beyond these specific proposals, the work performed in this WP6 also includes a diverse set of OpenAPIs, as described in Section 2.2.3.2.

- **Rec. 6**: It was recommended that Hexa-X-II could help to provide a sort of platform to the customers, so that, e.g., applications could be installed on the network (which is considered also as a way to monetise the networks).

  Answer:
  It is considered that the Decentralised Orchestration system part of the management framework presented in this document is well in line with this recommendation. Figure 3-65 in the previous Deliverable D6.3 [HEX224-D63] illustrates an example with an idealised representation of a GUI that could be used for deploying network services in line with such Decentralised Orchestration approach. The representation illustrates how different stakeholders (customers) can provide and use different software components in a sort of marketplace and use the 6G network as a platform where the composed network services can be deployed and executed, just as the recommendation suggests.

- **Rec. 7**: Recommended to consider GenAI-related techniques.

  Answer:
  The WP6 smart management framework is agnostic in what regards the application of specific AI/ML-based techniques, i.e., it does not require or preclude the application of specific techniques or algorithms. Although in certain cases it does use specific AI/ML algorithms, i.e. those from section 2.1.4, it is also designed modular and extendible with other AI/ML algorithms (or any other specific technique).
  In fact, the solution described in section 2.1.4.1 uses a Conditional Variational Autoencoder (CVAE) which is a type of Gen-AI model, used to generate Input-Output samples out of which the best one for that particular scenario is selected.
  However, other solutions from state-of-the-art GenAI techniques not explicitly developed in WP6 work could also be used with the framework proposed in this document. E.g., GenAI-based NLP techniques could be part of the intent-based management mechanisms to translate intents that could be expressed in natural language. Also, Generative Adversarial Networks (GAN) [GPM+14] could be used for generating synthetic data for training other AI algorithms in the framework (deep learning algorithms require large volumes of high-quality data, but in telecommunications, obtaining real data can be expensive or limited). In this context, GANs could be used to create realistic synthetic data, such as network traffic logs, to simulate dense traffic environments or extreme network conditions to train prediction and optimisation models. Also, Transformer-based networks [VSP+17] could be used in intelligent network traffic management by analysing large volumes of network traffic data in real time, enabling traffic prediction to anticipate peak demand in different parts of the network, predict infrastructure status changes (e.g., in the volatile extreme-edge domain), perform routing optimisation by dynamically adjusting resource allocation to avoid network congestions, or perform pattern analysis to identify anomalies or patterns in traffic, such as possible cyber-attacks or network failures.

4. Conclusions

This deliverable provides a fundamental contribution towards the design of the Hexa-X-II E2E 6G System Blueprint, presenting the design of the 6G Smart Network Management Framework. The detailed framework is the outcome of the work in the lifetime of WP6 in the HEXA-X-II work programme. It has been produced,

following the specification of a set of enablers to support smart network management functionalities, as detailed in D6.3 [HEX224-D63]. Based on the detailed enabling technologies, specific components have been derived and constitute part of the framework. The set of components are classified as overall M&O solutions, overall functionalities, specific systems and algorithms. Each one of the components has a specific role, while their synergy is required to support end-to-end network service management.

Upon the specification of the components of the 6G Smart Network Management Framework, their design and development are detailed. Implementation details are provided for the supported mechanisms and technologies per component. Following, a set of evaluation results are presented for individual components as well as workflows that consider the collaboration among multiple components. Part of these results is produced based on the work in progress in the PoCs. A set of KPIs are defined and considered in the evaluation.

In the provided 6G Smart Network Management Framework, a set of novel mechanisms and approaches are specified that will play dominant role in the evolution towards the 6G networks. Decentralised intelligence, automation, privacy and security characteristics are strongly considered and supported in the various management and orchestration mechanisms that are detailed. The exploitation of AI/ML technologies is included in most of the developments within WP6, while challenges related to the need for training and evaluation of the ML models to achieve the desired accuracy are identified. A variety of solutions for supporting orchestration actions is introduced based on the emergence of multi-agent approaches, cognitive control loops, decentralised schemes, and federation techniques. Their adoption and/or combination for tackling orchestration challenges for resources spanning across the network continuum is suggested.

With regard to the overall M&O solutions, current status and trends are identified for the multi-agent and decentralised orchestration approaches. It is noticed that multi-agent techniques are emerging to support deployment and management of network services over resources in the network continuum where different levels of synergy and collaborative actions may be established (e.g., agents acting under a single or multi-domain environment). Various trends are identified for the combination of multi-agent techniques with reliable and explainable AI frameworks to improve adoption of mechanisms, the emergence of agentic AI frameworks where multiple agents can collaborate and undertake different tasks (e.g., recommenders, RL-driven actions), the opportunity to combine these approaches with modern observability stacks (e.g., based on open telemetry specifications) to assist decision making by agents, the support of integration of network with far edge/edge/cloud technologies for time critical operations (e.g., real time data processing of high data volumes), and the need to examine trust, security and privacy aspects by the agents. The decentralised orchestration concept is also detailed, where future work many include the development of components to tackle scalability aspects, the integration of alternative AI/ML models for the ISPM component, and the examination of synergies with other M&O approaches.

With regard to the overall functionalities, current status and trends are also identified. In the case of monitoring and telemetry functionalities, it can be claimed that innovations in TeraFlowSDN, MEC integration, and automation frameworks pave the way for flexible, intelligent, and future-proof networks, aligning with emerging 6G requirements. Various enhancements are envisaged for the future, including the refinement of the supported AI/ML mechanisms for automation, and the promotion of the implementations toward standardisation. Contribution has been also provided to the zero-touch RT network automation functionalities, with descriptors for unified CL modelling, mechanisms and workflows for governance and coordination. A set of mechanisms have been implemented to support real-time CL functions and governance, combined with monitoring and telemetry functionalities, and multi-cluster resource orchestration mechanisms. In the case of the MCE, an Integration Fabric has been developed as inspired by ETSI ZSM, acting as a new integration/exposition based on event-driven approach. The detailed concept of the integration fabric can be generalised in the future, elevating as an active component to decentralise the way to coordinate network functions/component. An implementation is also provided for the SLA-driven federated orchestration functionalities, where improvements are planned in the future on prediction-based triggers of Smart Contracts for tighter latency bounds. In the case of trust management functionalities, a trust assessment function and an associated ontology have been designed for the network continuum, along with the definition of internal interfaces and data models between TEF and LoTAF. In the future, it is envisaged that the common trust management model will be able to share information across multiple domains, trust will be delivered as a transparent notary service (TNS) to promote external audits, trust level agreement will be declared to formalise end-user' requirements, while intent-based trust management will be supported.

The various specific systems and algorithms that are provided in the Smart Management Framework support various M&O operations, while lead to increased efficiency of the applied solutions. Novel systems and algorithms are introduced, while various ideas have arisen for the adoption and extension in the future. For instance, the NDT concept has been introduced to improve automation in network management and support optimal decision making by operators. The current work around the NDT can be considered as a starting point for advanced implementations in the future. Sustainable MLOps approaches are also very efficient in terms of sustainability requirements and are envisaged to be adopted a lot to serve ML workloads. Network programmability solutions are considered crucial, especially when combined with open APIs that support convergence with edge/cloud orchestration mechanisms (e.g., work produced around the CAMARA APIs) and are expected to be adopted and extended a lot in the future to support orchestration scenarios across the network continuum. Third-party resource control separation and user-centric service provisioning was examined, leading to solutions that extend Role-based Access Control (RBAC) mechanisms with dynamic, model-driven permissions tailored to tenants, by using dynamic URSPs (User Equipment Route Selection Policies) for personalised service activation, and the integration with closed-loop automation ensuring SLA compliance. This work is envisaged to be promoted within the 3GPP SA5 activities.

The provided outcomes from WP6 are directed to the various WPs of the HEXA-X-II work programme and mainly WP2, with a twofold objective. On the one hand, input is provided for the finalisation of the architectural blueprint of the 6G ecosystem. On the other hand, the development of management and orchestration mechanisms is provided as input to the PoCs under development in WP2, leading to a set of validation and evaluation results, reported through various KPIs. This work is considered as ongoing till the end of the lifetime of the HEXA-X-II work programme and will be reported in the upcoming Deliverable D2.6.

5. References

[28.538] 3GPP TS 28.538 version 17.3.0, "5G Management and orchestration; Edge Computing Management-", July 2023.

[28.809] 3GPP. (2021). TR 28.809: Study on enhancement of Management Data Analytics (Release 17). 3rd Generation Partnership Project. https://www.3gpp.org/

[28.831] 3GPP TR 28.831 version 18.0.0, "Study on basic Service-Based Management Architecture (SBMA) enabler enhancements", June 2023.

[29.520] 3GPP TS 29.520 version 15.3.0, "Network Data Analytics Services", April 2019.

[36.888] 3GPP TR 36.888, "Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE (Release 12)", June 2013.

[3GP21] 3GPP TR 21.905 – V18.0.0 – Vocabulary for 3GPP Specifications, Available at: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=558

[ABC23] k. Antevski, CJ. BC, "Applying Blockchain consensus mechanisms to Network Service Federation: Analysis and performance evaluation", Computer Networks, vol. 234, 2023

[Adl22] Arthur D. Little, "Operations 4.0 – The five tech pillars that will make or break telco operators in 2022", 2022, available at: https://www.adlittle.com/en/insights/report/operations-40 (last access Nov. 2024)[Cap24] Capgemini Research Institute, "Networks with Intelligence – Why and how the telecom sector should accelerate its autonomous networks journey", 2024, available at: https://www.capgemini.com/wp-content/uploads/2024/01/CRI_Autonomous-Network.pdf (last access Nov. 2024)

[CCK20] T. Chu, S. Chinchali, S. Katti (2020). Multi-agent Reinforcement Learning for Networked System Control. In Proceedings of the International Conference on Learning Representations (ICLR) 2020. doi: https://doi.org/10.48550/arXiv.2004.01339

[CDT18] T. Cerny, M. J. Donahoo, and M. Trnka, "Contextual understanding of microservice architecture: current and future directions", ACM SIGAPP Applied Computing Review, vol. 17, no. 4, pp. 29-45, 2018.

[CEC24] CAMARA Project. MEC Exposure and Experience Management API Specification. Retrieved from https://github.com/camaraproject/EdgeCloud/blob/main/documentation/Supporting

Documents/API%20proposals/Discovery/MEC%20exposure%20and%20experience%20management.yaml, 2024

[CLL+20] Chen, D., Lin, Y., Li, W., Li, P., Zhou, J. and Sun, X., 2020, April. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 04, pp. 3438-3445).

[DKJ18] A. Dorri, S. S. Kanhere and R. Jurdak, "Multi-Agent Systems: A Survey," in IEEE Access, vol. 6, pp. 28573-28593, 2018, doi: 10.1109/ACCESS.2018.2831228. keywords: {Task analysis;Multi-agent systems;Computer science;Security;Australia;Computational modeling;Decision making;Multi-agent systems;survey;MAS applications;challenges}

[DMIC23] Adamczyk, C., & Kliks, A. (2023). Detection and mitigation of indirect conflicts between xApps in Open Radio Access Networks. arXiv preprint arXiv:2305.13464.

[DSS04] M. Dorigo, T. Stu, and T. Sttzle. "Ant colony optimization." (2004).

[ERE22] Ericsson Blog, "Breaking the Energy Curve Report". Available online from https://www.ericsson.com/en/news/2022/10/ericsson-publishes-breaking-the-energy-curve-report-2022, 2022.

[ETS21] ETSI GR NFV 003 - V1.6.1 - Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV, Available at: https://www.etsi.org/deliver/etsi_gr/NFV/001_099/003/01.06.01_60/gr_nfv003v010601p.pdf

[ETS20] ETSI White Paper No. 40 Autonomous Networks, supporting tomorrow's ICT business 1st edition, October 2020.

[DZONE20] DZone, Real World Examples and Use Cases for Apache Kafka. Avaible at: https://dzone.com/articles/real-world-examples-and-use-cases-for-apache-kafka

[FAPI] https://fastapi.tiangolo.com/

[FGP+23] Ferriol-Galmés, M., Paillisse, J., Suárez-Varela, J., Rusek, K., Xiao, S., Shi, X., Cheng, X., Barlet-Ros, P. and Cabellos-Aparicio, A., 2023. RouteNet-Fermi: Network modeling with graph neural networks. IEEE/ACM transactions on networking, 31(6), pp.3080-3095.

[FSP+20] F. Faticanti, M. Savi, F. D. Pellegrini, P. Kochovski, V. Stankovski and D. Siracusa, "Deployment of Application Microservices in Multi-Domain Federated Fog Environments," 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 2020, pp. 1-6, doi: 10.1109/COINS49042.2020.9191379.

[FU98] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels", IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2384-2415, October 1998.

[GKV+19] T. Goethals, S. Kerkhove, L. Van Hoye, M. Sebrechts, F. De Turck and B. Volckaert, "FUSE : a microservice approach to cross-domain federation using docker containers." In V. M. Munoz, D. Ferguson, M. Helfert, & C. Pahl (Eds.), Closer: Proceedings of the 9th International Conference on Cloud Computing and Services Science, pp. 90–99, 2019. https://doi.org/10.5220/0007706000900099

[GFN] https://grafana.com/

[GPM+14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680. https://doi.org/10.48550/arXiv.1406.2661

[HEX24] Hexa-X project, https://hexa-x.eu/ (Accessed December 2024).

[HEX223-D22] Hexa-X-II, "Foundation of overall 6G system design and preliminary evaluation results", December 2023.

[HEX224-D23] Hexa-X-II, "Interim overall 6G system design", June 2024.

[HEX224-D24] Deliverable D2.4. End-to-end system evaluation results from the interim overall 6G system. Sept. 2024. [online] Available at: https://hexa-x-ii.eu/wp-content/uploads/2024/10/Hexa-X-II_D2_4_Final_2.pdf

[HEX223-D12] Hexa-X-II, "Deliverable D1.2: 6G Use Cases and Requirements", December 2023.

[HEX223-D32] Hexa-X-II, "Deliverable D3.2 Initial Architectural enablers", October 2023.

[HEX224-D33] Hexa-X-II, “Deliverable D3.3 Initial analysis of architectural enablers and framework”, April 2024.

[HEX223-D62] Hexa-X-II, “Foundations on 6G Smart Network Management Enablers”, October 2023.

[HEX224-D63] Hexa-X-II, “Initial Design of 6G Smart Network Management Framework”, June 2024.

[HS17] B. Hayes and J. A. Shah, “Improving Robot Controller Transparency Through Autonomous Policy Explanation,” in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, Vienna Austria: ACM, Mar. 2017, pp. 303–312. doi: 10.1145/2909824.3020233.

[ILE24] https://gitlab.com/decentralized-continuum-orchestration/infrastructure-layer-emulator

[ITU93] International Telecommunication Union (ITU), Maintenance: Introduction and General Principles of Maintenance and Maintenance Organization – Maintenance Terminology and Definitions, ITU-T Recommendation M.60. March 1993.

[ITU-Y.3057] ITU-T Y.3057, “A trust index model for information and communication technology infrastructures and services”, December 2021.

[JAN93] J.-S.R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665-685, May-June 1993, doi: 10.1109/21.256541

[JP09] J. Pearl, Causality. Cambridge university press, 2009.

[JTG+24] Jorquera Valero, J. M., Theodorou, V., Gil Pérez, M., &Martínez Pérez, G. (2024). SLA-driven trust and reputation management framework for 5G distributed service marketplaces. *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 1863-1875.

[K3S] K3S Lightweight Kubernetes [Online]. Available at: https://k3s.io (Accessed: 9 Oct. 2024).

[K8S] Kubernetes [Online]. Available at: https://kubernetes.io (Accessed: 9 Oct. 2024).

[KBF] https://www.kubeflow.org/docs/distributions/

[KBF+24] L. Karaçay, A:C. Baktir, R. Fuladi, E.D. Biyar, Ö.F. Tuna, and I. Arikan, 2024, September. Secure AI/ML-Based Control in Intent-Based Management System. In 2024 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 618-623). IEEE.

[KCK21] Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications 80(5):8091–8126

[KP] https://sustainable-computing.io/

[KLM22] Kokkonen, H., Lovén, L., Motlagh, N. H., Kumar, A., Partala, J., Nguyen, T., ... & Riekki, J. (2022). Autonomy and intelligence in the computing continuum: Challenges, enablers, and future directions for orchestration. arXiv preprint arXiv:2205.01423.

[Law24] A. Lawrence, “Good News – Growth For A Telco! Might Be Bad News For Some, Though…”, 6GWorld, September 2024. [Online]. Available at: https://www.6gworld.com/exclusives/growth-for-a-telco-might-be-bad-news-for-some-though/?utm_medium. Accessed: Oct. 2024.

[LHP+15] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

[LXD] LXD. [Online] Available at: https://documentation.ubuntu.com/lxd. Accessed: December 2024.

[MAJ+18] Mestres A, Alarcón E, Ji Y, Cabellos-Aparicio A. Understanding the modeling of computer network delays using neural networks. InProceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks 2018 Aug 7 (pp. 46-52)

[Maz75] J. E. Mazo, “Faster-than-Nyquist signaling”, Bell System Technical Journal, vol. 54, no. 8, pp. 1451-1462, October 1975.

[MCP+24] Miserez, J., Colle, D., Pickavet, M., & Tavernier, W. (2024). Exploiting Queue Information for Scalable Delay-Constrained Routing in Deterministic Networks. IEEE Transactions on Network and Service Management.

[MECP014] ETSI MEC PoC 014, “Network Resource Allocation”. Available at: https://mecwiki.etsi.org/index.php?title=PoC_14_Network_resource_allocation (Accessed May 2024).

[MIN] https://min.io/

[MNC+17] Muñoz, R., Nadal, L., Casellas, R., Moreolo, M.S., Vilalta, R., Fàbrega, J.M., Martínez, R., Mayoral, A. and Vílchez, F.J., 2017, June. The ADRENALINE testbed: An SDN/NFV packet/optical transport network and edge/core cloud platform for end-to-end 5G and IoT services. In 2017 European Conference on Networks and Communications (EuCNC) (pp. 1-5). IEEE.

[MOD11] S. Mitchell, M. Osullivan, and I. Dunning, “PuLP: a linear programming toolkit for python,” The University of Auckland, Auckland, New Zealand, vol. 65, 2011

[MQT] MQTT web page, https://mqtt.org/ (last access: Dec. 2024)

[NG13] Nadeau, Thomas D., and Ken Gray. SDN: Software Defined Networks: An authoritative review of network programmability technologies. " O'Reilly Media, Inc.", 2013.

[NOMAD] Nomad [Online]. Available at: https://www.nomadproject.io (Accessed: 9 Oct. 2024).

[OCM09] [Perry, G. (2009). The Open Cloud Manifesto: Version 1.0.9. Retrieved from https://gevaperry.typepad.com/Open%20Cloud%20Manifesto%20v1.0.9.pdf

[OGB18] F. K. Oduro-Gyimah and K. O. Boateng, "Application of CANFIS model in the prediction of multiple-input telecommunication network traffic," ITU Journal: ICT Discoveries, Special Issue No. 2, Nov. 2018, © International Telecommunication Union, 2018. [Online]. Available: https://www.itu.int/en/journal/002/Pages/default.aspx.

[Open API] https://www.openapis.org/

[OTL24] OpenTelemetry, available at: https://opentelemetry.io/

[PBM+24] R. Pires, H. Blue, J. Malinen, M. De Angelis, P.G. Giardina, G. Landi, M. Laukkanen, K. Aloha, and P. Porambage, "Closed-Loop Automation in 6G for Minimum Downtime Task Continuity in Surveillance Cobots", EuCNC and 6G Summit, June 2024.

[PC91] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs", Social Science Computer Review, vol. 9, no. 1, pp. 62-72, 1991.

[PFF+22] Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, *111*, 104755.

[PSQL] https://www.postgresql.org/

[PTC] https://pytorch.org

[PRE24] PREDICT-6G project, https://predict-6g.eu/ (Accessed April 2024).

[Prometheus] https://prometheus.io

[PV20] E. Puiutta and E. M. S. P. Veith, “Explainable Reinforcement Learning: A Survey,” in _Machine Learning and Knowledge Extraction_, vol. 12279, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds., in Lecture Notes in Computer Science, vol. 12279. , Cham: Springer International Publishing, 2020, pp. 77–95. doi: [10.1007/978-3-030-57321-8_5](https://doi.org/10.1007/978-3-030-57321-8_5).

[RET+24] F. Ruggeri, W. Emanuelsson, A. Terra, R. Inam, and K. H. Johansson, “Rollout-based Shapley Values for Explainable Cooperative Multi-Agent Reinforcement Learning,” in *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, Stockholm, Sweden: IEEE, May 2024, pp. 227–233. doi: 10.1109/ICMLCN59089.2024.10624777.

[RSS+18] T. Rashid. M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, S. Whiteson (2018). QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML) 2018.

[SAIN-23] RFC 9417: “Service Assurance for Intent-Based Networking Architecture”, July 2023.

[SBN+23] Sharma, P., Bhaskar, M., Ng, W., & Muraleedharan, A., "Why AI-powered RAN is an energy efficiency breakthrough". Ericsson. Retrieved from https://www.ericsson.com/en/blog/2023/1/ai-powered-ran-energy-efficiency, 2023.

[SCP] https://github.com/hubblo-org/scaphandre

[SQLA] https://www.sqlalchemy.org/

[SSB18] O. O. Semenova, A. O. Semenov, O. V. Bisikalo, P. I. Kulakov, R. R. Hamdi, R. Romaniuk, and B. Bissarinov, "Genetic ANFIS for scheduling in telecommunication networks," in Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, Wilga, Poland, 2018, vol. 10808, 108081Z. doi: 10.1117/12.2501503.

[STR15] Saito, Takaya, and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets." PloS one 10.3 (2015): e0118432.

[SWARM] Docker Swarm [Online]. Available at: https://docs.docker.com/engine/swarm (Accessed: 9 Oct. 2024).

[SVA+23] J. Suárez-Varela *et al*., "Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities," in IEEE Network, vol. 37, no. 3, pp. 146-153, May/June 2023, doi: 10.1109/MNET.123.2100773

[TAZ+13] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services". IEEE Communications Magazine, vol. 51, no. 8, pp. 155-161, August 2013.

[TAZ+22] I. Tzanettis, C-M. Androna, A. Zafeiropoulos, E. Fotopoulou, S. Papavassiliou. Data Fusion of Observability Signals for Assisting Orchestration of Distributed Applications. *Sensors*. 2022; 22(5):2061. https://doi.org/10.3390/s22052061

[TF] https://www.tensorflow.org/

[TFS] ETSI TeraFlowSDN, https://tfs.etsi.org

[TFS24] ETSI TeraFlowSDN contributions from HEXA-X-II, https://labs.etsi.org/rep/search?group_id=96&project_id=74&repository_ref=master&scope=issues&search=HEXA-X-II

[TIB+20] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick, and E. Fersman, "Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan: IEEE, Dec. 2020, pp. 1–7. doi: 10.1109/GLOBECOM42002.2020.9322496.

[TIF22] A. Terra, R. Inam, and E. Fersman, "BEERL: Both Ends Explanations for Reinforcement Learning," *Applied Sciences*, vol. 12, no. 21, p. 10947, Oct. 2022, doi: 10.3390/app122110947.

[TDB24] https://docs.timescale.com/

[TFS24] ETSI TeraFlowSDN SDG, https://tfs.etsi.org/ (Accessed April 2024).

[TKC23] R. Talebi, A. Khamseh, and M. Cheraghali, "Fuzzy analysis of the influence of factors on the integration of telecommunication technology infrastructure using ANFIS," Fuzzy Optimization and Modeling Journal, vol. 4, no. 1, Apr. 2023, doi: 10.30495/fomj.2023.1988350.1093.

[TS] https://pytorch.org/serve/

[VMC+21] Vilalta R, Muñoz R, Casellas R, Martínez R, López V, de Dios OG, Pastor A, Katsikas GP, Klaedtke F, Monti P, Mozo A. Teraflow: Secured autonomic traffic management for a tera of sdn flows. In2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) 2021 Jun 8 (pp. 377-382). IEEE.

[VHT+24] K. Vandikas, H. Hallberg, S. Ickin, C. Nyström, E. Sanders, O. Gorbatov, and L. Eleftheriadis, "Ensuring energy-efficient networks with artificial intelligence", Ericsson Technology Review, April 2022, available at: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/ensuring-energy-efficient-networks-with-ai (Accessed April 2024)

[VVG+24] R. Vilalta, F. J. Vílchez, Ll. Gifre, C. Manso, J.L. Carcel-Cervera, R. Leira, J. Aracil-Rico, J.P. Fernández-Palacios, R. Martínez, R. Casellas, R. Muñoz, Providing Anomalous Behaviour Profiling by extending SmartNIC Transceiver support in Packet-Optical Networks, OFC, 2024.

[VSP+17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

[WDB+18] J. van der Waa, J. van Diggelen, K. van den Bosch, and M. Neerincx, "Contrastive Explanations for Reinforcement Learning in terms of Expected Consequences," *IJCAI/ECAI Workshop on explainable artificial intelligence*, Jul. 2018, Accessed: Apr. 13, 2022. [Online]. Available: http://arxiv.org/abs/1807.08706

[Yeg11] Yegge, S. (2011). Stevey's Google Platforms Rant. [Online]. Available: https://gist.github.com/chitchcock/1281611 (Accessed: Dec. 2024)

[YLC+21] L. Yang, Y. Lu, J. Cao, J. Huang and M. Zhang, "E-Tree Learning: A Novel Decentralized Model Learning Framework for Edge AI," in IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11290-11304, July 2021, doi: 10.1109/JIOT.2021.3052195

[YLO24] https://yolov8.com

[YMP23] S. Yrjölä, M. Matinmikko-Blue and P. Ahokangas, "Developing 6G Visions with Stakeholder Analysis of 6G Ecosystem," *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Gothenburg, Sweden, 2023, pp. 705-710, doi: 10.1109/EuCNC/6GSummit58263.2023.10188379.

[YRH20] H. Yau, C. Russell, and S. Hadfield, "What did you think would happen? Explaining agent behaviour through intended outcomes," in *Advances in neural information processing systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 18375–18386. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/d5ab8dc7ef67ca92e41d730982c5c602-Paper.pdf

[ZFF+24] A. Zafeiropoulos, N. Filinis, E. Fotopoulou and S. Papavassiliou, "AI-Assisted Synergetic Orchestration Mechanisms for Autoscaling in Computing Continuum Systems," in IEEE Communications Magazine, 2024, doi: 10.1109/MCOM.001.2200583.

[ZFV+23] Zafeiropoulos, A., et al. (2023). "Intent-Driven Distributed Applications Management Over Compute and Network Resources in the Computing Continuum." *19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, Pafos, Cyprus, pp. 429-436. DOI: 10.1109/DCOSS-IoT58021.2023.00074.

[ZSM-009-1] ETSI GS ZSM 009-01, "Zero-touch network and Service Management (ZSM); Closed loop Automation; Part 1: Enablers", v1.1.1. June 2021.

[ZSM-009-2] ETSI GS ZSM 009-02, "Zero-touch network and Service Management (ZSM); Closed loop Automation; Part 2: Solutions for automation of E2E service and network management use cases", v1.1.1. June 2022.

[ZSM-009-3] ETSI GS ZSM 009-03, "Zero-touch network and Service Management (ZSM); Closed loop Automation; Part 3: Advanced topics", v1.1.1. August 2023.

[ZSM-002] ETSI GS ZSM 002-01, "Zero-touch network and Service Management (ZSM); Reference Architecture", v1.1.1. August 2019

[ZSM-003] ETSI GS ZSM 003, "Zero-touch network and Service Management (ZSM);End-to-end management and orchestration of network slicing, v1.1.1, June 2021

6. Annexes

This Annexes section, includes information regarding certain updates performed on the technical enables already described in the previous Deliverable D6.3 [HEX223-D63] (in annexes A.1, A.2, and A.3). Besides, it also provides information regarding KPIs (annex A.4).

A.1 Level of Trust Assessment Function

The Level of Trust Assessment Function (LoTAF) focuses on the development of synergetic mechanisms for the assessment of trustworthiness in the Network Continuum. In particular, LoTAF intends to reduce the uncertainty inherent in multi-stakeholder and cross-domain contexts [JTG+24] by evaluating security aspects of network services within 6G networks. Therefore, the Trust Management System functionality proposes LoTAF together with Trust Evaluation Function (TEF) [HEX223-D63] as two main functions to evaluate the Level of Trust or Trust Index, respectively, of infrastructure components, compute nodes, services, and 3rd party consumers.

Design Principles

The Level of Trust (LoT) concept is one of the foundations of the Hexa-X-II aiming to address the KVI of trustworthiness. To this end, LoTAF facilitates the management of selecting distributed resources and infrastructure in 6G-oriented scenarios, e.g., Network Continuum. It functions as an impartial, two-way service that evaluates the degree of trustworthiness that can be placed in network services before their deployment and usage, while also assessing the first LoT during the operation of an end-to-end connection. Hence, LoTAF assists trustors (i.e., users) in making informed decisions and trustees (i.e., network providers) with insights regarding adherence to trust requirements previously offered and opportunities for improving service quality.

LoTAF is geared toward ensuring end-to-end trustworthiness in Network Continuum, spanning multiple domains and providers. It aligns with established standards like ITU-T Y.3057 [ITU-Y.3057] for managing trust across ICT infrastructures and services. Furthermore, LoTAF aims to support service assurance in intent-based networking (IBN) [SAIN-23], providing quantifiable attributes necessary for trust evaluation. LoT management is divided into two primary blocks: (i) semi-natural trust-based intent and (ii) the Level of Trust Assessment Function (see Figure 6-1).

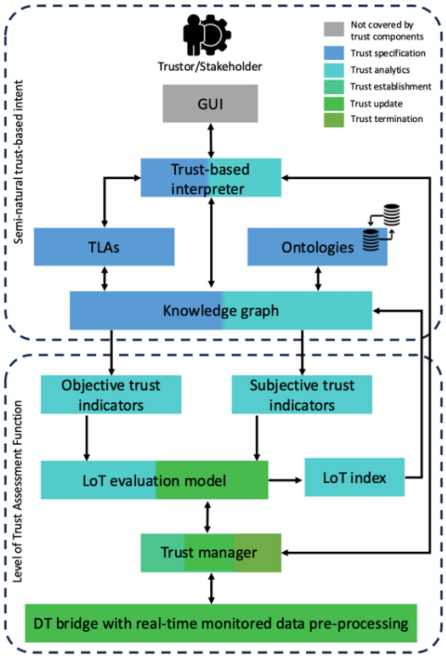

Figure 6-1: (Figure 6-1) High-level overview of LoT Functionalities

The first block focuses on semantic processing, mapping, and the representation of trust requirements. End-users engage with this entry-point block via a Graphical User Interface (GUI), where they articulate their trust

needs in natural language. A trust-based interpreter is employed to analyse these inputs and identify the key elements necessary for determining which network services meet these requirements. To ensure the proper comprehension of trust requirements, an ad-hoc Named Entity Recognition (NER) model aids in this process by extracting relevant tokens. To validate the results from our NER model, the trust-based interpreter may employ two different methods for ensuring consistency in trust intent: a) a one-touch approach, where a chatbot assists the end-user and simplifies the process of capturing the required tokens, or b) a zero-touch approach, which reduces user interactions by applying reinforcement learning techniques to comprehend all tokens from a single input and only confirm minor details with end-users. At this stage, the network service catalogue is explored to identify possible options that meet the trust requirements. It is important to note that these trust requirements can be integrated with existing intents to carry out several pre-filtering steps. Following this, the system matches the trust requirements with appropriate network services from a catalogue. Once a selection is made, a Trust Level Agreement (TLA) is formalised between the involved parties.

As a result of this block, a knowledge graph is created, incorporating the user's intent and the TLA details. Since no existing trust-driven ontology for Network Continuum was found, a custom ontology was developed using the Linked Open Term [PFF+22] approach, considered in ETSI SAREF standards. This knowledge graph serves as a repository for events and data concerning the health and symptoms of network services, aiding in the evaluation of trustworthy KPIs and service adherence to the TLA.

The second block emphasises monitoring, ensuring, and forecasting any deviations from the Level of Trust established in the TLA. The LoTAF assesses both objective indicators (quantifiable metrics) derived from the RFC 9417 theoretical framework [SAIN-23] and subjective indicators (non-technical factors) based on third-party recommendations, personal experience, or reputation. Initially, both dimensions are used to assess the available network services, i.e., the analysis of a set of available network services in a catalogue, but once a trust relationship is established, only objective indicators are utilised for continuous LoT updates. Besides, Bayesian methods are then applied to determine the degree of alignment between the LoT and predefined thresholds outlined in the TLA. It is important to emphasise that this process is not a single occurrence; rather, it is executed on an ongoing basis throughout the entire duration of the business relationship, using time-windows for regular assessments.

As can be observed in Figure 6-2, the components and functionalities described in the above paragraphs are associated with different colours that, in turn, represent the lifecycle phases of the Level of Trust. Concretely, LoTAF has been inspired by the ITU-T Y.3057 [ITU-Y.3057], although two new phases, trust specification and trust analytics, have been designed to cover the whole LoT lifecycle.

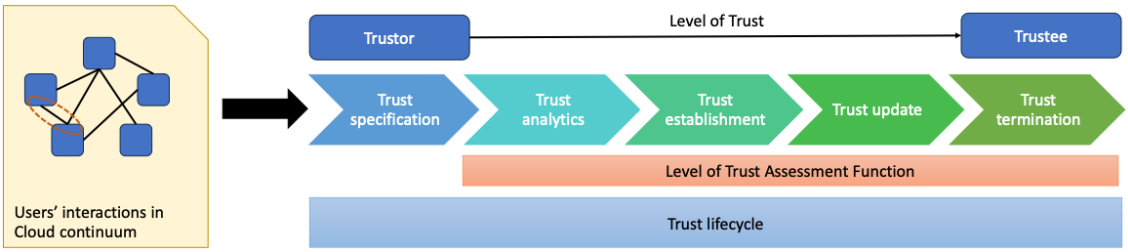

Figure 6-2: Level of Trust lifecycle phases.

In general, the LoT lifecycle phases are driven by the two main participants in the LoTAF: the trustor, who is the entity that places trust or relies on another party (consumer), and the trustee, who is the entity that is trusted or evaluated for trustworthiness (network provider). Thus, the LoT lifecycle is divided into five phases:

- *Trust specification*: trust requirements and criteria are specified by end-users and the LoTAF needs to look into the catalogue of available network services for candidates that can meet those terms. Therefore, it sets the expectations for trust based on security and reliability factors to be evaluated in later phases. Besides, this phase also entails the beginning of a TLA between trustor and trustee.
- *Trust analytics*: data and symptoms are collected and analysed to assess the component's ability to meet the specified trust criteria at the beginning of a relationship but also once an establishment needs to be updated in real time. Objective and subjective indicators are used to evaluate trustworthiness, and the data are stored in the knowledge graph to infer new information in forthcoming requests.

- *Trust establishment*: after the initial LoT is determined, a trust relationship is formalised between the trustor and trustee through the Trust Manager entity. Furthermore, trust level agreements or policies are settled between trustor and trustee.
- *Trust update*: since trust is not a one-time process, LoTAF needs to continually monitor and reassess throughout Trust Manager and information sources new data or symptoms. In this vein, trust levels are adjusted to reflect ongoing compliance with the established trust requirements.
- *Trust termination*: when a trust relationship is no longer needed or the component no longer meets the required trust requirements, Trust Manager leads to the end of the trust relationship and the deactivation of the whole trust lifecycle.

In summary, LoTAF offers a novel and dynamic method to improve the security and trustworthiness of 6G network services, providing both users and service providers with valuable insights and supporting better decision-making.

Workflows

Figure 6-3 outlines a sequence of interactions involving the Level of Trust Assessment Function within a trust-based intent management scenario.

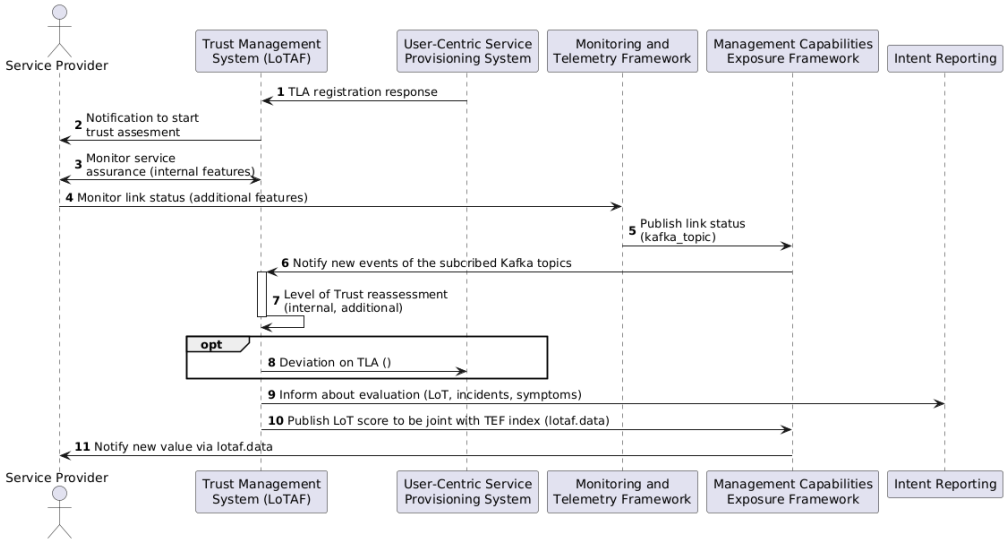

Figure 6-3: Level of Trust Assessment Function workflow for evaluation agreed trust requirements.

The diagram begins with the User-Centric Service Provisioning System (UCSPS) when a new business relationship is going to commence between multiple stakeholders (both Service Provider(s) and Service Consumer). In this regard, the UCSPS shares the Trust Level Agreement (TLA) settled by all parties where trust attributes are declared. The Trust Management System is composed of two subcomponents such as the Level of Trust Assessment Function (LoTAF) and the Trust Evaluation Function (TEF). Nevertheless, this section is only related to the LoTAF. At this point, the LoTAF interprets trust intents to notify the Service Provider of trust requirements to be ensured during a whole relationship. As a result, the monitoring process starts at the Service provider to check the assurance. On the one hand, the LoTAF deploys its own monitoring engine which is based on the RFC 9417 theoretical framework and subjective features coming from recommendations or historical feedback. On another hand, the Monitoring and Telemetry Framework complements an additional set of features to be considered to update the LoT or identify deviations. So as to get them, the LoTAF is subscribed to the proper Kafka topic of Management Capability Exposure (MCE) in which Monitoring and Telemetry Framework forwards data. Afterwards, LoTAF reassesses the initial value of LoT based on real-time information and finds out potential conflicts that may entail breaking the TLA. Whether a deviation on the contracted TLA is found, a notification will be sent to the User Centric Service Provisioning System. Next, LoTAF informs the new trust evaluation to the Intent Reporting module as trust is contemplated as a new intent that can be requested during the Intent-Based Management process. Then, LoTAF publishes the LoT with the TEF module, the other subcomponent of the Trust Management System, to merge two trust values into a unique one. Such a sharing process is performed through the Management Capability Exposure because LoTAF and TEF make use of Integration Fabric as a communication bus to share

information. Last but not least, LoTAF communicates the new LoT to both the Service Provider and the Service Consumer who are subscribed to LoTAF Kafka topic in the MCE.

A.2 Open Telemetry and Data Fusion

For supporting management and orchestration of 6G services and infrastructures, a data observability framework is implemented covering the heterogeneous needs of the systems developed to maintain and optimise operations. Support for various types of data is considered, coming from both the infrastructure and the executed services. Thus, elements like computing and network nodes, extreme-edge devices, synchronous and asynchronous communication, are modelled and monitored to provide detailed inputs in a manner in which they can be analysed and generate valuable insights. Such data can follow multiple formats, but three are distinctive enough [TAZ+22] to analyse further:

- Metrics: Timeseries values representing specific variables such as latency, throughput, resource consumption.
- Logs: Text outputs originating in entities of the infrastructure (e.g. computing, network nodes, deployed services etc.) that can be analysed to identify regular and irregular behaviour.
- Traces: Services tend to become more and more modular and so, they form call chains that can be traced hop-to-hop to find hidden relations and identify performance issues.

For handling such heterogeneous data sources, the OpenTelemetry [OTL24] framework provides support with the developed instrumentation libraries, but also offers the OTLP Collector which tackles data collection, aggregation and preprocessing. Specifically, it offers a homogeneous method to adopt ready-to-use or implement custom data *receivers*, *processors* and *exporters* that fit the needs of specific infrastructure or services. To offer flexibility, receivers and exporters can implement both push and pull modes for supporting both active and passive monitoring in both ends of the collector.

These different modes are described as:

- Push mode receivers: This mode requires active data sources that push data at real-time whenever they are generated or in batches.
- Pull mode receivers: Probing-based, passive sources that require activity from the collector to provide the collected data (e.g. collecting data from REST APIs)
- Push mode exporters: This mode requires the collector to actively access the passive consumers and push the collected data. For an external component to obtain the data at real-time this mode is necessary.
- Pull mode exporters: This mode sets up passive exporters that are accessed by active external components (e.g. Prometheus)

For deploying the collector, a unified configuration is provided listing the following:

- Data sources (receivers): Components providing data by pushing it or
- Processing functions (processors)
- Data storage/analysis (exporters)

An example is given in Figure 6-4.

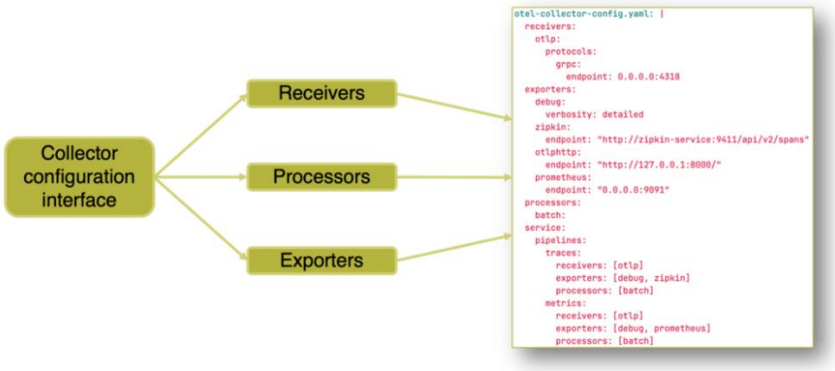

Figure 6-4: Indicative collector configuration.

The different stages of observability data collection and storage are displayed in Figure 6-5.

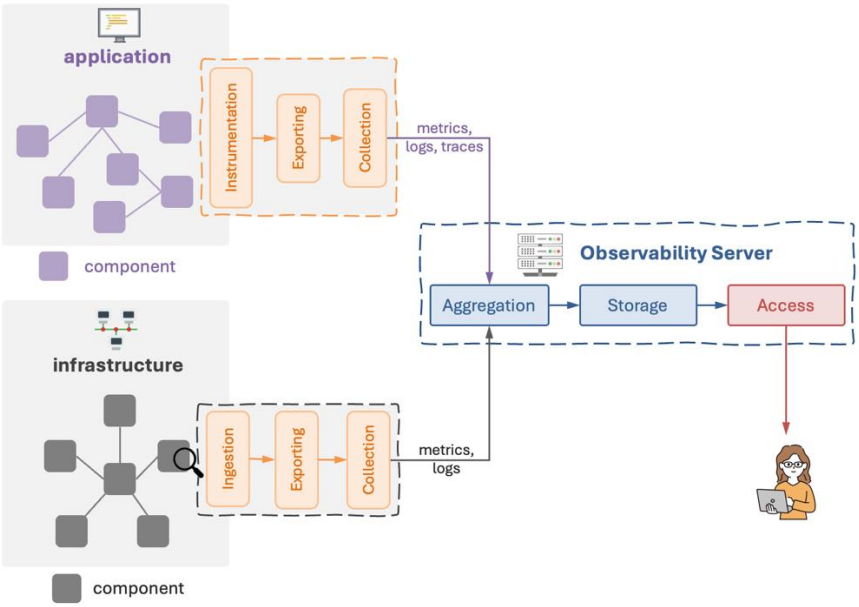

Figure 6-5: Overview of observability data collection and storage stages.

Close to the application and the infrastructure, data is homogenised and collected by the collector. In the case of the application, instrumentation is required to expose the data before exporting and collection, while for the infrastructure, data is ingested using off-the-shelf monitoring tools that expose the corresponding data. At the server side, data is aggregated by the different sources and stored at a dedicated infrastructure that also offers the corresponding access to external components.

Telemetry data collection over the network

When data is collected across the network a more specific approach is used. Collection points across different nodes act in a decentralised manner, collecting data locally. Taking network overhead into consideration, data is optimally transferred across the network to central locations and finally at the server. For implementing this approach, an agent-gateway method is used, where agents handle local data collection and forward the collected datasets to gateways residing at local locations as shown in Figure 6-6.

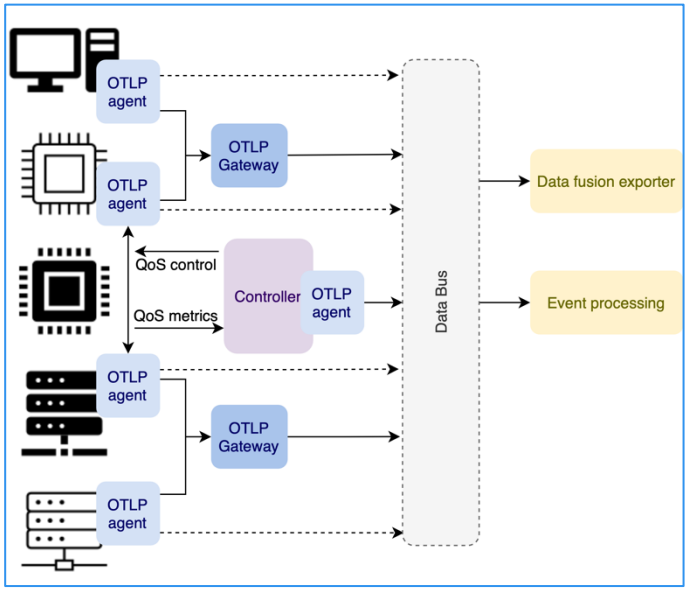

Figure 6-6: Telemetry data collection over the network.

Data fusion

Observability data is typically collected and stored in disaggregated storage spaces from where it can be accessed individually. The ability to aggregate and process heterogeneous data in a fused manner can unlock hidden connections that can majorly influence decision-making and enhance issue identification. Metrics can be used for monitoring basic aspects of the network, but for locating specific issues appearing traces and logs can offer a more user-centric view of the infrastructure and services executing on it. This is not possible after the collection of the data, since the information to correctly merge the service data with the underlying infrastructure is not available. Thus, a data model to sufficiently identify connections between the different data types is necessary at the instrumentation level. In Figure 6-7, an example of a trace and its spans, enhanced with the corresponding metrics and logs is displayed.


```
"7258791432255": {
    "object-detector-span": {
        "start": "2024-07-25 10:45:24.179842",
        "end": "2024-07-25 10:45:24.710385",
        "pod": "object-detector-c7f598568-8xsc2",
        "logs": [
            {
                "timestamp": "2024-07-25 10:45:24.710293",
                "message": "Detected objects: ['chair', 'chair', 'ch
            },
            {
                "timestamp": "2024-07-25 10:45:24.179948",
                "message": "Received Frame ID: 7258791432255\nTimest
1721893524179841848"
            }
        ],
        "metrics": [
            {
                "energy": "10.909090909090908"
            }
        ]
    },
    "frame-sampler-span": {
        "start": "2024-07-25 10:45:24.139563",
        "end": "2024-07-25 10:45:24.178545",
        "pod": "sampler-sender-54f54587c5-xh8nc",
        "logs": [
            {
                "timestamp": "2024-07-25 10:45:24.179000",
                "message": "Frame ID: 7258791432255\nTimestamp 1: 17
            },
            {
                "timestamp": "2024-07-25 10:45:24.139650",
                "message": "Frame received from source\nFrame ID: 72
            }
        ],
        "metrics": [
            {
                "energy": "0.32727272727272727"
            }
        ]
    }
```


Figure 6-7: Indicative example of fused trace, spans, logs, and metrics.

In this case, Kubernetes was used as a deployment platform, thus, the corresponding pod information was used to identify the computing nodes where the service components were executed and obtain the corresponding metrics and logs.

A.3 RT zero-touch cognitive closed-loop automation

This annex intends to give more information on RT zero-touch cognitive closed loops automation described in Sec. 2.2.1.3. Closed-loop automation is a dynamic process that monitors, adjusts, and controls systems to meet predefined objectives, commonly used in networks like 5G. By incorporating causal reasoning into this process, the system can go beyond reactive adjustments and make intelligent, informed decisions based on cause-and-effect relationships. Causal reasoning enables the identification of root causes, proactive interventions, and the simulation of potential scenarios, which enhance the system's ability to optimise performance, improve stability, and prevent issues before they arise. This integration ultimately leads to more adaptive, resilient, and intelligent autonomous systems capable of handling complex and evolving environments. With a causal graph, all causal relation between action-KPIs and KPIs-KPIs are known, this will bring important benefits as we mention below:

- **Enhanced Root Cause Analysis:** By using a causal graph, the root cause of a networking issue can be better identified as all causal relations are known. There might be many root-causes to a particular problem, and using a causal graph can help identifying the most probable causes and directing us to the most relevant correction actions.
- **Improved Explainability:** When we observe a networking issue (e.g., QoE degradation), we can explain why the problem arises (e.g., QoE degraded because throughput decreases but latency looks

good). This capability enables the capability of explainability, and allows the operator to explain why a specific action is taken but not the others (e.g., allocate more bandwidth since we observe the reason is throughput degradation)

- **Continuous Improvement for Optimisation:** The proposed solution allows to perform optimum actions to obtain the desired KPI value based on the causal relations and functions, which are continuously improved and validated.
- **Self-adaptiveness:** The solution is useful in scenarios where the system is nonstationary in terms of its causal structures. This capability helps to update the causal graph whenever actual system's causal relation changes.

Causality BackgroundFigure 6-8 considers five variables, also referred to as nodes, X, Y, Z, W, T. An arrow (also called directed edge or link) indicates there is causal relation between two given nodes. For example, the arrow is pointed from X to Z, which means that X is a cause of Z or X is causing Z. As X is causing Z, there are also dependent nodes. In other words, when we know X we can give information about Z. There is also a relation between W and T but this relation is not causal. When we have data for W and T and analyse the data, we can see this relation, but this is not a causal relation rather a correlation**.**

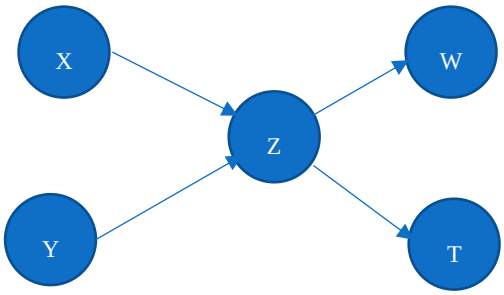

Figure 6-8: An example of a Causal Graph.

Suppose a 5G network experiences performance degradation, such as increased latency or reduced throughput. The network uses machine learning models that rely on correlations, which may only suggest likely causes without certainty (e.g., a rise in user demand may correlate with increased latency, but it could be due to a deeper issue like poor load balancing).

Causal Approach: Causal models help identify the root causes of performance degradation by distinguishing between confounding factors and actual causes. For instance, a causal graph might show that a specific configuration change (e.g., a modified scheduling algorithm) led to increased latency, despite the rise in user demand.

Closed-Loop Action: Once the cause (configuration issue) is identified, the system can autonomously roll back the change or adjust the configuration based on causal insights.

In the following, it is explained how the estimated causal graph is utilised for the operation of autonomous management of multiple CLs.

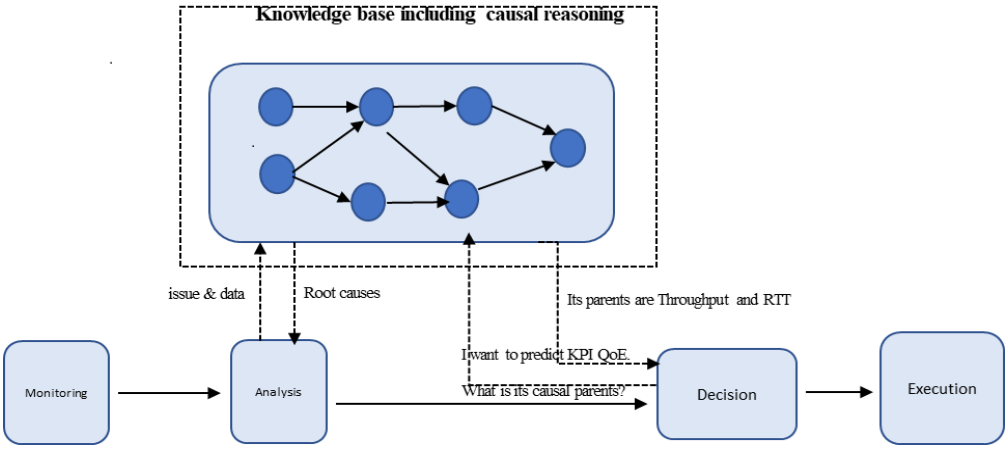

Figure 6-9: CLs with Causal graph.

Figure 6-9 shows how a causal graph can be used with a CL. The causal reasoning with a causal graph can be utilised for Analysis and Decision module of a CL operation. The causal relations among the actions and KPIs can be effectively used at Analysis module in order to find out the root cause leading to an issue (i.e., unmet KPI expectation or fault). When an issue is raised due to a performance degradation (e.g., latency becomes unmet), the corresponding set of Decisions that are capable of proposing actions to mitigate this issue are triggered. Similar to the typical fault management procedures, the Decision first needs to find the root cause that raised the issue. In this regard, the Decision can subscribe to the causal graph service provided by the knowledge base with causal information.

A.4. Key Performance Indicators

In this Annex section, the main KPIs considered for each of the framework's components are described. For facilitating reading, several explanatory notes have been added at the end of the section and are referenced by the KPI descriptions when needed. In case a KPI is considered out of scope, the cell is left blank.

Table 6-1: KPIs for Overall M&O Solutions and Functionalities.

KPI	Multi-agent system for multi-cluster orchestration	Decentralised orchestration	Monitoring and telemetry	RT zero-touch CLs automation & coordination	SLA-driven Federated Orchestration	Trust Management	Management capabilities exposure
Scalability	Inference/Response/Training time across number of agents/services/clusters	Provisioning time for new infrastructure nodes (considering: Initialisation and configuration time, connection time, and warm-up time) / Provisioning cost (see Note 1)	Number of concurrent monitored parameters and devices and open telemetry streams.	Number of concurrent closed loops that the system can manage.	Number of provider domains that can be connected to the DLT network	Number of trust requests that system can properly manage during the resource management and network service provisioning	Kafka metrics: -throughput, -partition distribution, -consumer lag
Latency	End to end latency to serve a service request	Service latency measurements in deployments involving deployments on extreme-edge nodes.	Service latency at run time.	Service latency at runtime, also considering service components at extreme-edge nodes.	Round Trip Time of federated service		-E2E latency communication from message production to consumption. - Latency related to the onboarding of a new component
Flexibility	SLA violations under dynamic workloads	N.A. As described in D6.3, this is more a "quality" than a "performance indicator".	Number of supported monitoring and telemetry protocols.	Qualitative indicator; capability of the system to manage and coordinate different	Ease of operation in the LCM of federated service		

				types of closed loops.			
Services Creation Time	Service (re)deployment time	The time it takes for a network service to be deployed by the M&O system.		Time required to instantiate and configure closed loop functions during the service provisioning phase.	Time taken to federate service		
Reliability	Request success rate	Reliability measures the system ability to function consistently and without failure over a specific period. There are many metrics that can be used to measure reliability. See Note 2.	Mean Time To Detect.	Service failure rate; recovery time.		Times that the system's compute nodes succeeded to execute a task/workload within a time threshold	-Consumer Lag During Failures (track consumer lag if brokers fail), -replication count, -Producer and consumer error rate, -Metrics of data persistence stored in Redis
Programmability		Possible programmability metrics in Note 3.		API surface and completeness. Ease of integration.			
Automation		The following specific metrics could be used related to automation: - Automation coverage. - Time savings - Reduction in human errors - MTTR - Operational costs savings - Service delivery time - Services uptime (reliability) Note 4 for further details	Mean Time To Detect.	Time required to complete a full closed loop cycle, from monitoring to execution.			

Availability	Percentage of time the service is operational			Service continuity.	Continuity of service in provider domain	Percentage of time the system's compute nodes were not fully loaded or unavailable.	-Uptime Percentage, -Mean Time to Recovery (MTTR), -Incident Response Time
Maintainability		Here the metrics that could be used to measure maintainability: - Mean time to repair (MTTR) - Fault Detection Rate - Frequency of Maintenance Activities - Preventive vs. Corrective Maintenance Ratio - Change Failure Rate - Time to Implement Changes - Self-Healing Efficiency - Network Downtime Due to Maintenance - Troubleshooting Success Rate More details in the attached Note 5.					
Security							- Mean Time to Resolution (MTTR), -Manual Intervention Frequency, -Automated Recovery Success Rate

Processing Capacity	Requests served per second	The following metrics can be used - Number of computing devices - Processing power of computing devices (e.g., CPUs, ASICs, FPGAS...) to perform operations. Can be measured in operations per second (e.g., MIPS, GFLOPS). - The processing capacity itself can be also a metric - see attached Note 6					
Integrated Intelligence		See Note 7.		Qualitative indicator; possibility to adopt AI/ML techniques in the various stages of the closed loops.			
Intent Expressiveness		See Note 8.					
OPEX	Power consumption	The OPEX is a metric in itself. However, it can be computed in different ways. See Note 9.		Reduction of human interventions; reduction of power consumption; reduction of resource utilisation.			
Energy Efficiency	Energy gain compared to centralised deployments			Reduction of power consumption through specialised closed loops.			
User Experience Data Rate				Increased user experience data rate (through CLs for service migration or			

				adaptation of resource allocation)			

Table 6-2: KPIs for Specific Systems.

KPI	Network programmability	Third-party resource control separation	User-centric service provisioning	Network Digital Twins Creation	Sustainable MLOps	Secure AI/ML-based control for Intent-based Management
Scalability	Number of network domains. Number of controlled network devices.			Number of represented infrastructure/network entities		
Latency				Model accuracy (distance between predicted and measured latency)		
Flexibility	Supported network domains and control protocols. Supported network device types and protocols.					
Services Creation Time	Connectivity service latency at run time.					
Reliability		Rate of isolated failures within stakeholder environments versus total system failures. Mean time Between failures (MTBF) for isolated stakeholder systems.	Mean Time Between Failures (MTBF). Error and failure rate analysis. Effectiveness of error recovery mechanisms (e.g., SLA compliance metrics).			

		Mean time to Recovery (MTTR) after a stakeholder-specific failure.				
Automation	MTTD. MTTR				The following metrics could be measured: - Automation level. - Efficiency of Automated Decisions. (S-MLOps Note 1)	
Availability		Service uptime percentage measured over time. Frequency and duration of downtime incidents caused by resource contention. Resource utilisation fairness index across stakeholders.	Service uptime percentage. Frequency and duration of downtime incidents. Incident recovery time using closed-loop automation.			
Maintainability		Number of updates or patches applied without affecting other stakeholders. Time taken to isolate and resolve issues within a stakeholder's environment. Ease of applying configuration changes using predefined boundaries.	Time taken for updates and scaling without service disruption. Mean Time to Repair (MTTR) for issues. Automation level in applying updates using defined SLA-based service assurance workflows.			
Security		Number of unauthorised access attempts detected and mitigated. Compliance with security policies and				Success rate of adversarial attacks and enhancement of the model robustness against such attacks.

		audits. Incident response time to security breaches. Penetration testing results specific to stakeholder resource isolation.				
Processing Capacity				Inference time on specified hardware		
Energy Efficiency					Energy Monitoring Capability Index (EMCI) (S-MLOps Note 2)	

Table 6-3: KPIs for Algorithms.

KPI	**ML based configuration recommender for energy savings**	**Efficient network and service function allocation**	**Resource assignment for federated learning**	**Multi-agent RL for adaptive scaling**	**Explainability for RL-based Control**
Scalability		Number of properly served service requests	Percentage of network edge data resources engaged	Inference/Response/Training time across number of agents/services/clusters	
Latency		End-to-end latency of the deployed services	Total time taken between training request to trained model response	End to end latency to serve a service request	
Flexibility		Amount of SLA violations for services due to network dynamics		SLA violations under dynamic workloads	
Services Creation Time				Service (re)deployment time	
Reliability		Packet loss	Percentage of trained models meeting the minimum required performance thresholds	Request success rate	

Automation	Automatic adjustment of network configurations with respect to the change in network and requirements	Response to network dynamics (migration, scaling)		Performance gain compared to static thresholds scenarios	
Availability				Percentage of time the service is operational	
Elasticity				Elasticity efficiency (time required for scaling, performance gain compared to BAU scenario)	
Processing Capacity		Number of service requests served		Requests served per second	
OPEX		Power consumption		Power consumption	
Energy Efficiency	Energy reduction compared to static-rule based system	Number of servers used by the deployed services (consolidation)	The number of exchanges between nodes weighted by the network tier in which they occur before model convergence to requested level	Energy gain compared to static deployments	
Explainability					Accuracy and fidelity of explanations

Note 1 - Scalability

Horizontal scalability (i.e., the ability to add more nodes or instances to the system without affecting performance or functionality) is considered crucial here. In this regard, the following metrics are considered relevant:

- Provisioning Time for New Nodes: This metric measures how long it takes to provision and activate a new node, considering the total time elapsed from the initiation of the process to add a new node until it becomes fully operational. Effective horizontal scalability implies short provisioning times so that new nodes can become functional quickly.
  This metric can be broken down by considering different provisioning stages, for example:
  - Initialisation and Configuration Time: The time needed to install software, load initial data, and configure the node to operate within the system.
  - Connection Time: The time it takes for the new node to integrate into the network and begin operating.
  - Warm-Up Time. This measures the node's initial performance once it is operational, which may be lower than its stable performance until it fully loads data and configures necessary connections. Minimizing warm-up time ensures that the new node can handle load at maximum performance quickly.
- Provisioning Cost: In addition to time metrics, it can also be relevant to evaluate the additional cost involved in adding new nodes, which can help to understand the relationship between cost and the benefits of rapid scaling.

Note 2 - Reliability

Metrics that can be used to measure reliability:

- *Mean Time Between Failures (MTBF)*
  Description: Measures the average operational time before a failure occurs. It is one of the most common reliability metrics and represents the system's ability to operate continuously.
  Calculation: MTBF = Total operating time / Number of failures.
  Example: If a machine operates for 1,000 hours and has 5 failures, its MTBF is 200 hours.
- *Mean Time to Repair (MTTR)*
  Description: Measures the average time required to repair a system or component after a failure. A lower MTTR indicates quicker recovery, improving reliability.
  Calculation: MTTR = Total repair time / Number of failures.
  Example: If a machine has 5 failures and the total repair time is 10 hours, the MTTR is 2 hours.
- *Failure Rate*
  Description: The probability of a failure occurring over a specific period.
  Calculation: Failure Rate = Number of failures / Total operating time.
  Example: If an application fails 3 times in 1,000 hours of operation, its failure rate is 0.003 failures per hour.
- *Availability*
  Description: Measures the percentage of time a system or component is operational and ready for use. Availability combines MTBF and MTTR to calculate overall system reliability.
  Calculation: Availability = MTBF / (MTBF + MTTR).
  Example: If a system has an MTBF of 200 hours and an MTTR of 2 hours, its availability is approximately 99%.
- *Mean Time Between Maintenance (MTBM)*
  Description: Measures the average time between preventive and corrective maintenance activities. A higher MTBM suggests the system requires less maintenance and is thus more reliable.
  Calculation: MTBM = Total operating time / Number of maintenance events.
- *Corrective Maintenance Rate*
  Description: Measures the number of times a system requires corrective maintenance within a specific timeframe.
  Calculation: Corrective Maintenance Rate = Number of corrective maintenance events / Total operating time.
  Example: If a system needs 5 corrective maintenances in 1,000 hours, its corrective maintenance rate is 0.005 per hour.
- *Recovery Time*
  Description: Measures the total time required to restore the system to its operational state after a failure. Short recovery time indicates high reliability.
  Example: In IT services, this could measure the time from a system disruption to full operational status.
- *Failure Density*
  Description: Measures the number of failures per unit of time or use, such as "failures per operating hour"
  Example: If a car component fails twice every 10,000 kilometres, its failure density is 0.0002 failures per kilometre.
- *Critical Incident Rate*
  Description: Measures the frequency of severe or critical failures that directly impact operation. This metric is particularly useful in systems where some failures are minor and others critical.
  Example: In a software application, the critical incident rate can be measured as the number of crashes or severe errors per thousand hours of use.
- *Component Reliability Index*
  Description: Measures the reliability of individual components within a system, allowing the identification of less reliable components to improve the overall system reliability.
  Example: In an electrical network, the reliability index for each transformer can be measured and analysed to identify the network's most vulnerable points.
- *Durability*

Description: Measures the amount of time or volume of operation that a system or component can withstand before failure. Although often associated with material resilience, it is a critical metric in any reliability assessment.
Example: For an engine, durability might be measured in terms of operational hours before a major overhaul is required.

Note 3 - Programmability

Below are some metrics that can be considered to measure *programmability*:

- *API Surface and Completeness*
  Description: Measures the scope and coverage of the APIs available for programming the system. This includes the number of functions, classes, methods, and events exposed for customisation and automation.
  Calculation: Number of endpoints, functions, or methods available in the API compared to the total number of functions in the system.
  Example: If an API exposes 80% of the core system functionalities, it is highly programmable.
- *Ease of Integration*
  Description: Evaluates the time and effort required to integrate the system with other services or systems through programming. This can be measured by the average time required to complete common integrations.
  Calculation: Average time to implement common integrations / Number of integrations.
  Example: In a system with a well-documented API, the integration time might be only a few hours, while a less programmable system might take days or weeks.
- *Customisation Development Time*
  Description: Measures the time a developer needs to create or modify specific functionalities in the system.
  Calculation: Average time to implement a common customisation / Number of customisations implemented.
  Example: A system with easy-to-use development tools and good documentation will allow customisations to be made in less time compared to a more rigid system that takes longer.
- *Extensibility Rate*
  Description: Measures the system's ability to accept new functionalities through extensions, modules, or plugins. This can be measured by the number of extensions added over a period of time or the percentage of external functionalities that can be added.
  Example: In programmable systems, the number of modules developed and integrated in the last year can be measured to evaluate its extensibility.
- *Documentation and Support Quality*
  Description: Measures the quality and completeness of the system's documentation, API, SDK, and development tools, as good documentation facilitates programmability. It may also include technical support and the availability of forums or communities.
  Calculation: Documentation quality (via user surveys) + Technical query resolution rate / Average response time.
  Example: A system with clear and complete documentation makes it easier for developers to work with and allows for quicker and more effective use of programming tools.
- *Number of Supported Programming Languages and Tools*
  Description: Evaluates the number and variety of programming languages and tools supported for customizing or programming the system.
  Example: A system that allows programming in multiple languages (like Python, JavaScript, and C#) and supports multiple IDEs (Integrated Development Environments) is more programmable than one that only allows a specific language or tool.
- *Mean Script Deployment Time*
  Description: Measures the average time required to implement and run scripts or automations in the system. Shorter times indicate greater programmability.
  Example: In a system where scripts can be quickly tested and deployed, the deployment time is short, encouraging the creation of automations.
- *Backward Compatibility and Code Reusability*
  Description: Evaluates how easy it is to update the system without needing to rewrite previous customisations. The ability to reuse code in new system versions is an indicator of a good architecture for programmability.
  Calculation: Number of customisations that don't need modification when updating the system / Total number of customisations.
  Example: In systems with high version compatibility, customised code continues to work after updates, saving time and improving programmability.
- *Testing and Debugging Support*
  Description: Measures the ability to test and debug customised code within the system, as good testing and debugging support enhances programmability.
  Example: Systems that allow automated testing and offer debugging tools are more programmable because developers can test and adjust their code more efficiently.
- *Adaptability Rate*
  Description: Measures the system's ability to adapt to new functionalities or integrations without requiring a major rewrite of existing code.
  Calculation: Number of new functionalities adapted without restructuring / Total number of new functionalities added.
  Example: In a modular system, new features can be integrated without needing to redo the system's code architecture, which indicates high adaptability.

Note 4

- *Automation coverage*
  Definition: The percentage of network operations that are automated versus those still performed manually.

Formula:

$$\text{Automation Coverage (\%)} = \left(\frac{\text{Automated Tasks}}{\text{Total Tasks}}\right) \times 100$$

- *Time Savings*
  Definition: Reduction in time to execute a task due to automation.
  Formula:

$$\text{Time Savings (\%)} = \left(1 - \frac{\text{Automated Task Execution Time}}{\text{Manual Task Execution Time}}\right) \times 100$$

  Example Use: Quantifying how much faster automated processes like fault resolution or network provisioning are compared to manual operations.
- *Reduction in Human Errors*
  Definition: The decrease in the number of errors due to manual interventions as a result of automation.
  Formula:

$$\text{Error Reduction (\%)} = \left(\frac{\text{Manual Errors} - \text{Automated Errors}}{\text{Manual Errors}}\right) \times 100$$

  Example Use: Demonstrating improved reliability in routine network operations.
- *MTTR (Mean Time to Repair)*
  Definition: The average time to detect, diagnose, and resolve a network issue.
  Impact of Automation: Lower MTTR due to quicker anomaly detection and automated resolution mechanisms.
- *Operational Cost Savings*
  Definition: The reduction in operational expenditure (OPEX) due to automation.
  Formula:

$$\text{Cost Savings (\%)} = \left(1 - \frac{\text{Automated OPEX}}{\text{Manual OPEX}}\right) \times 100$$

  Example Use: Highlighting the financial benefits of automating network maintenance or provisioning.
- *Service Delivery Time*
  Definition: The time taken to provision or deploy a service.
  Impact of Automation: Significant reductions in service delivery time due to automated workflows for provisioning.
- *Service Uptime (Reliability)*
  Definition: Measurement of service availability and its consistency.
  Impact of Automation: Automation minimises downtime by proactively identifying and resolving issues.

Note 5

- *Mean Time to Repair (MTTR)*
  Definition: The average time required to repair a failed component or restore a system to operational status.
  Formula:

$$MTTR = \frac{\text{Total Repair Time}}{\text{Number of Repairs}}$$

  Relevance: Indicates how quickly issues can be resolved.
- *Fault Detection Rate (FDR)*
  Definition: The percentage of faults or issues detected before they cause significant impact.
  Formula:

$$FDR = \frac{\text{Number of Faults Detected}}{\text{Total Faults Occurred}} \times 100$$

  Relevance: Higher rates suggest better monitoring and diagnostic capabilities.
- *Frequency of Maintenance Activities*
  Metric that quantifies how often maintenance tasks (both preventive and corrective) are performed within a specified time period. It helps evaluate the balance between maintaining system reliability and avoiding excessive maintenance efforts or costs.
  Formula:

$$\text{Frequency of Maintenance Activities (FMA)} = \frac{\text{Number of Maintenance Activities}}{\text{Time Period}}$$

- *Ratio of Preventive to Corrective Maintenance*
  Definition: The proportion of preventive maintenance tasks compared to corrective tasks.
  Formula:

$$\text{Ratio} = \frac{\text{Number of Preventive Maintenance Tasks}}{\text{Number of Corrective Maintenance Tasks}}$$

  Relevance: A higher ratio suggests that preventive strategies are effective in reducing unexpected failures.
- *Change Failure Rate (CFR)*
  Measures the proportion of changes made to a system or network (e.g., upgrades, patches, or configuration updates) that result in unexpected issues, failures, or require rollback.
  Formula:

$$\text{Change Failure Rate (CFR)} = \frac{\text{Number of Failed Changes}}{\text{Total Number of Changes}} \times 100$$

- *Time to Implement Changes (TIC)*
  Measures the average time required to execute a planned change in the network, from initiation to completion.
  Formula:
  $$\text{Time to Implement Changes (TIC)} = \frac{\text{Total Time Spent on Changes}}{\text{Number of Changes Implemented}}$$
  Efficiency Indicator: Shorter implementation times indicate streamlined processes and effective change management.
  Operational Impact: Long implementation times can increase downtime risks and disrupt critical operations.
  Helps identify bottlenecks in planning, testing, or execution of network changes.
- *Self-Healing Efficiency (SHE)*
  The Self-Healing Efficiency measures the percentage of issues or failures in a network that are resolved automatically without human intervention.
  Formula:
  $$\text{Self-Healing Efficiency (SHE)} = \frac{\text{Number of Issues Resolved Automatically}}{\text{Total Number of Issues Detected}} \times 100$$
  Automation Benchmark: Higher SHE indicates the effectiveness of automated systems in maintaining network reliability.
  Cost and Time Savings: Reduces the need for manual intervention, minimizing operational expenses and response times.
  Scalability Indicator: Essential for large-scale networks where manual maintenance may not be feasible.
- *Network Downtime Due to Maintenance (NDM)*
  The Network Downtime Due to Maintenance measures the total amount of time the network is unavailable due to planned or unplanned maintenance activities.
  Formula:
  Network Downtime Due to Maintenance (NDM) = Sum of Downtime During Maintenance Activities (hours)
  Reliability Metric: Lower NDM values reflect better planning and execution of maintenance with minimal disruption.
  User Impact: Prolonged downtime can negatively affect user satisfaction and business operations.
  Optimisation Insight: Helps organisations optimise scheduling and methods for maintenance tasks.
- *Troubleshooting Success Rate (TSR)*
  Measures the percentage of troubleshooting efforts that successfully identify and resolve network issues on the first attempt.
  Formula:
  $$\text{Troubleshooting Success Rate (TSR)} = \frac{\text{Number of Successful Troubleshooting Cases}}{\text{Total Number of Troubleshooting Attempts}} \times 100$$
  Effectiveness Indicator: A high TSR reflects skilled personnel, robust diagnostic tools, and effective troubleshooting methodologies.
  Time and Resource Efficiency: Higher success rates reduce downtime and maintenance costs.
  Training and Tools Assessment: Identifies gaps in skills or tools that can hinder maintenance efficiency.

Note 6 – Processing Capacity

The Processing Capacity itself is a metric to measures the maximum amount of data or transactions that a network, device, or system can efficiently process per unit of time.

Different formulas can be used:

- For general data - in Bits per second (bps), Megabits per second (Mbps), Gigabits per second (Gbps), etc:
  $$\text{Processing Capacity} = \frac{\text{Number of Packets Processed}}{\text{Time Unit (seconds)}}$$
- For network packets - packets per second (pps):
  $$\text{Processing Capacity} = \frac{\text{Number of Transactions}}{\text{Time Unit (seconds)}}$$
- For requests or transactions (transactions per second - tps):
  $$\text{Processing Capacity} = \frac{\text{Data Processed (bits or bytes)}}{\text{Time Unit (seconds)}}$$

Relevance:
1. Rel. to. Network Scalability: Helps determine if the infrastructure can handle increases in traffic or load without degrading performance.
2. Bottleneck Identification: Reveals limitations in hardware or network configuration that affect processing efficiency.
3. Capacity Planning: Assists in decision-making for investments and upgrades to ensure the network meets future demands.
4. Equipment Comparison: Allows for evaluating network devices (routers, switches, servers) in terms of performance.

Note 7

Integrated intelligence refers to the network's ability to automatically adapt, optimise, and respond to changing conditions, using advanced technologies like AI/ML and network automation. Here key metrics that can help assessing the degree of intelligence integrated into a network:

- *Self-Healing Efficiency (SHE)*

The Self-Healing Efficiency measures the percentage of issues or failures in a network that are resolved automatically without human intervention.
Formula:

$$\text{Self-Healing Efficiency (SHE)} = \frac{\text{Number of Issues Resolved Automatically}}{\text{Total Number of Issues Detected}} \times 100$$

Relevance: A higher self-healing efficiency suggests the network's capacity to manage disruptions autonomously, which enhances reliability and reduces downtime.

- *Forecasting Accuracy*
  Definition: Measures the accuracy with which a network can predict future patterns (e.g., traffic patterns, devices availability patterns, resource usage patterns...) using historical data and machine learning models.
  Relevance: Accurate forecasting allows the network to proactively adjust its capacity, routes, or resources to meet future demands, optimizing performance and preventing congestion.
- *Anomaly Detection Rate*
  Definition: Measures the ability of the network to identify unusual or unexpected patterns in network traffic, performance, or behaviour that might indicate a problem, security threat, or system failure.
  Relevance: A high anomaly detection rate indicates that the network can recognise and react to problems before they escalate, allowing for quicker responses to issues.
- *Autonomous Resource Allocation*
  Definition: Measures how well the network can automatically allocate or adjust resources (such as bandwidth, computing power, or storage) based on real-time network demands.
  Relevance: An intelligent network can optimise resource allocation dynamically, ensuring that critical services always have the necessary resources while preventing resource wastage.
- *Fault Detection and Resolution Time*
  Definition: Measures the average time it takes for the network to detect, diagnose, and resolve faults automatically.
  Relevance: A network with integrated intelligence reduces the time to detect and fix problems, leading to higher availability and performance.

Note 8 - Intent

- *Intent Mapping Precision*
  Measures how accurately and reliably the system translates high-level user intents into network configurations and policies that align with the intended outcomes.
  Relevance: A high precision score means that the system can consistently convert abstract intent into practical, actionable network behaviour without unintended consequences or misconfigurations.
  Metric Example: Percent of intents successfully mapped to correct network configurations without manual intervention or error.
- *Intent Validation Success Rate*
  Measures the success rate of validating that the network can indeed meet the expressed intent (i.e., feasibility of the intent), especially when it comes to conflicting or complex network requirements.
  Relevance: This metric indicates the system's ability to identify conflicts, errors, or infeasibilities in the intent before they are implemented in the network.
  Metric Example: Percentage of intents that pass validation checks for conflict resolution or feasibility assessments without requiring manual adjustments.
- *Intent Error Rate*
  Measures the rate at which the network fails to meet the expressed intent due to misinterpretation or misconfiguration of the translated policies.
  Relevance: A lower error rate indicates that the IBN system can accurately interpret and implement user intent without frequent errors, ensuring better network performance and fewer disruptions.
  Metric Example: Number of incidents where the intent was not successfully implemented, divided by the total number of intents expressed.

Note 9 - OPEX

- *Total OPEX (Overall Operational Expenses)*
  The total sum of all costs associated with the operation and maintenance of the network, excluding capital expenditures (CAPEX). This includes expenses like staffing, network maintenance, utilities, licences, and third-party services.
  Formula: Total OPEX=Personnel Costs+Maintenance and Support+Software Licences+Utilities+Third-Party Services
  Relevance: Provides an overall snapshot of the financial health and efficiency of a telecom operator's operations. It is often expressed on a monthly or yearly basis to track trends and manage budgets.
- *OPEX per Customer (OPEX/C)*
  Measures the operational costs incurred for each customer or user served by the network. This metric helps assess the efficiency of delivering services to customers and can help identify areas for cost optimisation.
  Formula: OPEX per Customer = Total OPEX / Number of Active Customers
  Relevance: A key metric for understanding the cost of service delivery and can be used to compare different customer segments. A lower OPEX per customer indicates more efficient operations.
- *Network Maintenance OPEX (Maintenance Costs)*
  Refers to the expenses related to maintaining the network infrastructure, such as repair costs, preventive maintenance, and spare parts. This also includes the cost of network monitoring, testing, and upgrades.

Formula:

$$\text{Network Maintenance OPEX} = \text{Cost of Repairs} + \text{Cost of Preventive Maintenance} + \text{Cost of Spare Parts} + \text{Cost of Monitoring and Testing}$$

Relevance: Allows network stakeholders to assess the effectiveness and efficiency of their network maintenance activities. High maintenance costs may signal the need for infrastructure upgrades or process improvements.

- *Power and Energy Costs (OPEX related to Energy)*
  Telecom industry is energy-intensive, and electricity is a major cost component of OPEX. This metric measures the cost of power and energy required to run network equipment, including data centres, base stations, and network hubs.
  Formula: Energy OPEX = Energy Consumption × Energy Price per Unit
  Relevance: Network stakeholders can use this metric to identify energy inefficiencies and opportunities for cost savings through power optimisation strategies (e.g., network optimisation, renewable energy adoption, energy-efficient equipment).
- *Cost per Fault or Incident (OPEX related to Fault Management)*
  Measures the operational cost incurred per network fault or service incident. It helps operators understand the financial impact of service disruptions and identify areas for improvement in fault management processes.
  Formula: Cost per Incident = Total OPEX related to Faults / Number of Faults or Incidents
  Relevance: Indicates the effectiveness of the network in preventing or managing service disruptions. High costs per incident may suggest that the network is inefficient in handling faults and requires better fault isolation or automation.
- *Service Support OPEX (Customer Support and Service Costs)*
  Measures the cost associated with providing customer support and managing service inquiries, complaints, and technical support. This includes call centre operations, technical assistance, and customer care services.
  Formula: Support OPEX = Customer Support Salaries + Call Centre Costs + Service Platform Costs
  Relevance: Helps operators understand the cost of supporting their customer base and provides insights into the efficiency of their customer service processes. It also highlights areas where automation (e.g., chatbots, AI-driven support) could reduce costs.
- *OPEX Reduction from Automation*
  Measures the reduction in operational expenses due to the implementation of automation technologies, such as self-healing networks, automated provisioning, and AI-driven network management.
  Formula:

$$\text{OPEX Reduction from Automation} = \frac{\text{OPEX before Automation} - \text{OPEX after Automation}}{\text{OPEX before Automation}} \times 100$$

  Relevance: This metric highlights the impact of automation on reducing costs, improving efficiency, and freeing up resources for other investments.

Note 10

- *Automation Level*
  Definition: Measures the proportion of M&O processes and decisions handled automatically instead of manually.
  Formula:

$$\text{Automation Level (\%)} = \frac{\text{Automated Processes}}{\text{Total Processes}} \times 100$$

  Relevance: Indicates the system's overall automation maturity and progress toward zero-touch operations.
- *Efficiency of Automated Decisions*
  Definition: Evaluates the effectiveness of automated decisions in terms of error reduction and response time.
  Formula:

$$\text{Success Rate (\%)} = \frac{\text{Successful Automated Decisions}}{\text{Total Automated Decisions}} \times 100$$

  Relevance: Validates the quality and reliability of automated decision-making.

Note 11

- Energy Monitoring Capability Index (EMCI)
  Definition: Measures the system's ability to accurately track and monitor energy consumption across different stages of MLOps workflows (e.g., training, inference, data preparation). This includes criteria such as the granularity of measurements, real-time monitoring capabilities, and coverage of components involved in the workflows.
  Relevance: Evaluate the robustness and reliability of the system in tracking energy usage. It enables the identification of inefficiencies and supports the optimisation of machine learning operations with an emphasis on sustainability and energy efficiency.
  Formula:

$$\text{EMCI} = \left(\frac{\text{Points Achieved}}{\text{Total Points Available}} \right) \times 100$$